UNIVERSITY OF OSLO
Department of informatics

Analysis of service identification in SOA methodologies - with a unification in POSI, Perspective Oriented Service Identification

Master thesis

60 credits

Geir Anders Nilsen

<8 December 2008>

Contents

List of Tables

List of Figures

Abstract

This thesis is written in the context of Model Driven Architectures and SOA, and investigates different methodologies and their ways of identifying and describing a service oriented architecture. Methodologies analyzed are OASIS, ARIS, COMET-S and Archimate. With the knowledge from the analysis a perspective oriented service identification methodology will be proposed- POSI. POSI will be analyzed and later compared with the thesis initial analysis of mentioned methodologies.

Part I
Introduction to POSI

1 Introduction

This thesis is written in collaboration with the ongoing SHAPE project at SINTEF. The project is lead by Arne J. Berre whitch is also has been the main supervisor. He has been a really good supervisor, thanks to A.J. I would also like to thank my family for support in these last years, and Erik Hagen for help with modelling tools.

2 The Thesis

This thesis is about how to identify key requirements of a SOA and do an analysis of a SOA(Service Oriented Architecture) methodology. Further to extract the different the techniques to identify the services. The thesis is investigating that these different techniques can be composed in a certain order, to identify services with respect to the different service perspectives. The thesis will also define the various service perspectives.

3 Problem domain and context

As the title communicates the thesis is written in the context of SOA. Service Oriented Architectures or SOA is a really loaded term and is widely used and probably also misused. Many will argue that producers of software development technologies has hijacked the term to refer to their integration or development tools rather than an architectural methodology. Does this way of thinking fundamentally miss the point of services? Meaning SOA is not only a technology but a way of thinking flexible organizations that technology enables. Devotees of the latter argue that technology is an integrated part of the concept SOA and that technology and humans are intertwined. We are

mutually affected by each other in a heterogeneous web of actors[1], humans and non-humans. Technology should therefore be included as an actor or participant from the very start in the creation of services and other SOA artifacts. The objective of a service should according to the current methodology be representing what the business does, and define the boundary's to and from all actors. It is this that should be the focus in the creation of a service architecture. Technology and humans should be modelled side by side as participants or actors.

4 Problem definition

The SOA concept has clearly become a design paradigm. The adoption of SOA takes the decoupling of monolithic applications by decomposing business functions and processes into discrete reusable services. Further SOA is dealing with frequent and unpredictable changes by constructing an architectural model, discipline and abstraction that loosely couples service providers from service consumers. In this paper we take a look at the fundamental principles around design of SOA and how it can help the development of systems that addresses agility and adaptability- the adaptability to respond to changes and new requirements. According to some IBM's[2]experiences from early SOA implementation, projects show us that the existing development methodologies, processes and notations such as OOAD (Object Oriented Analysis and Design), EA (Enterprise Architectures) , and BPM (Business Process Management)comes to short in describing the requirements needed to support the SOA paradigm. It would be interesting to investigate what is missing and how these services should be identified out of its raw form into a perfectly shaped service crystal. The thesis are as mentioned above in the context of SOA. In the model world that would mean that this paper will be positioned in the area of CIM, PIM and the way from the first to the latter. I will try to evaluate each and everyone of 4 different approaches to Service oriented methodology's against some given requirements which will be expanded on later. The different approaches to evaluate is the ARIS [3], COMET-S [4], OASIS SOA works. OASIS SOA works are a collection of documents including the methodology[5], reference model[6] and reference architecture[7]. At the end the analysis may uncover how to identify business ideas into manageable services. Some of the challenges to be met are first how to design systems that provide a good fit between business processes, business goals and IT-architectures. It is obviously that a main activity is to identify services in such a methodology. An other important aspect is to ensure that the relationship between business processes and services is preserved and trailable. Secondly how to build architectures that are able to quickly respond to future changes.

5 Methodology and description of method of work

So how can the different methodologies be evaluated and classified to reflect their inherent properties? The method used in this thesis takes a set of requirements fitted to measure the properties of the methodology. The methodologies analyzed is chosen because they have different approaches for identifying and describing a SOA. The methodologies will be analyzed according to the set of requirements highlighting parts that will be brought into the POSI methodology. POSI will ten be exemplified and evaluated according to the same set of requirements used to evaluate the state of the art technologies. The requirements will be described in part 13, and will cover areas outlined by a analysis framework. As a guide line the following analysis framework will be used seen in figure 1. Then in the conclusion the questions raised in the thesis will be addressed and other findings described. In the end ideas for further work are proposed.

First the notation and language which are divided into CIM (computer independent models) , PIM (platform independent model) and the PSM (platform specific models). Secondly the Process which includes discipline, phases, roles and project management. Thirdly and last the tools to support the two first areas which includes enterprise modelling tools, system modelling tools and integrated development environments.

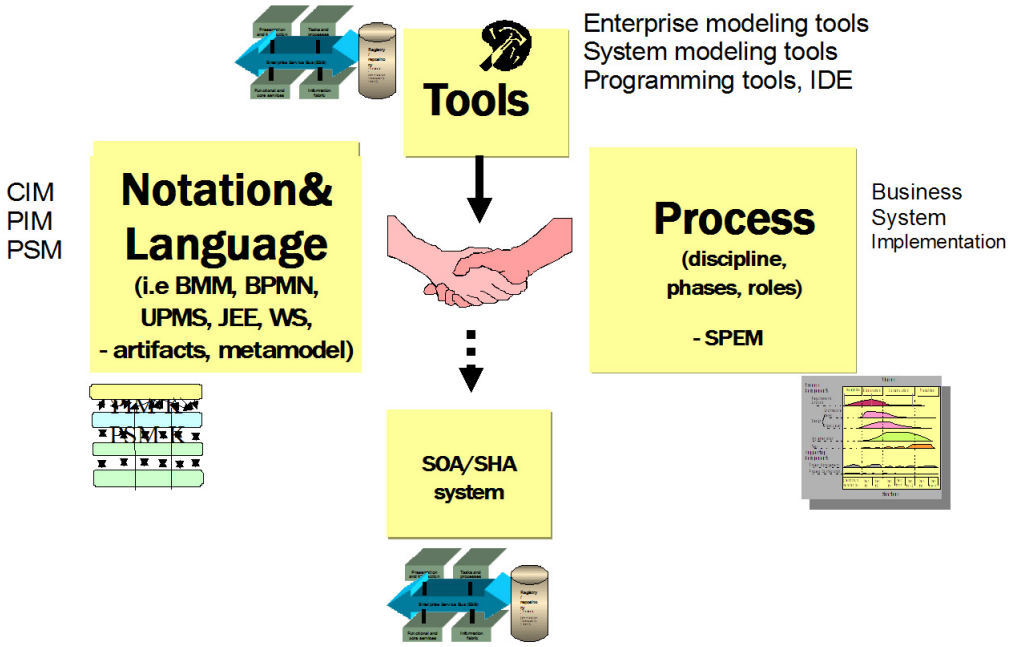


Figure 1: Analysis reference model

5.1 Processes

By process we mean the practical roadmap from start to the end, if there is any. This is kind of the project management view, and aims to plan the project with respect to resources and their roles, phases, delivery's and their milestones. In this lies also the different disciplines like requirements, analysis, design, implementation and test. One example of such is the figure 2 below, that shows a development process.

5.2 Notation and language

The notation and language is representing how, and what of the architecture, that can be formally represented according to the different layers of abstraction like CIM,PIM or PSM. Examples of these are BPMN, BMM, UML profiles and meta models for BPEL and UPMS and many more. At these layers there exists several areas as subject of investigation. Zachman has provided a useful

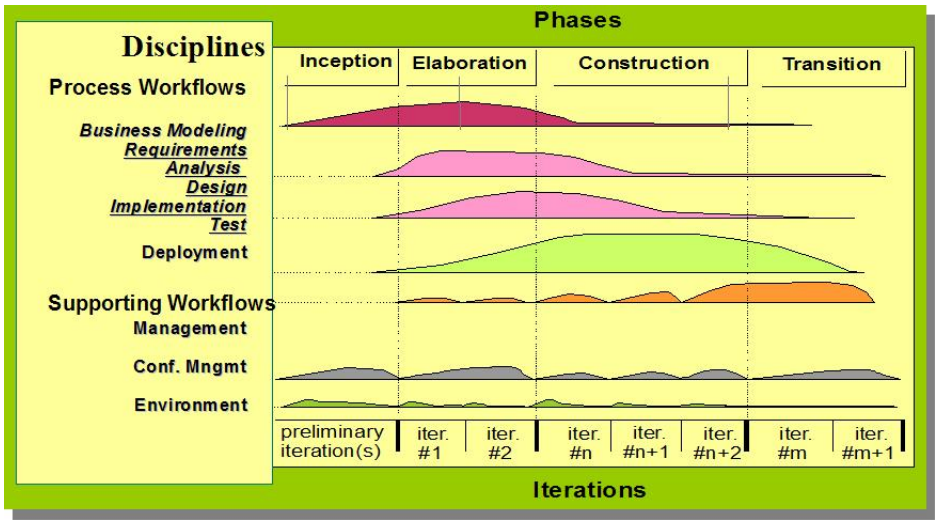


Figure 2: Process

framework for describing or viewing the architecture of an information system. The Zachman framework describes architecture from the perspective of the various stakeholders (rows in the matrix) and focuses on different aspects of architecture(columns in matrix), see figure 77 in the appendix XVI.

The matrix used in this reference architecture is based on the Zachman framework but the rows are altered to fit the different levels of the architecture(CIM, PIM and PSM). The columns are also adapted to fit with the different aspects of architecture. These aspects are ontology or information, process, business rules, goals, Non Functional aspects and qualities and last the organizations. With inspiration from the Zachman framework[8] and the aspects from A.J Berre's presentation at Semantic Days in Stavanger [9], There has been identified these aspects to use in the reference architecture(figure 3).

- Ontologies -what concepts are used to express the challenges in the domain and their relations.
- Goals -The goals that the business seeks to realize and are used to identify success or failure.
- Business Services -The services that realize the goals (Need-capability's)
- Business processes -what end to end processes are necessary to realize the business goals.

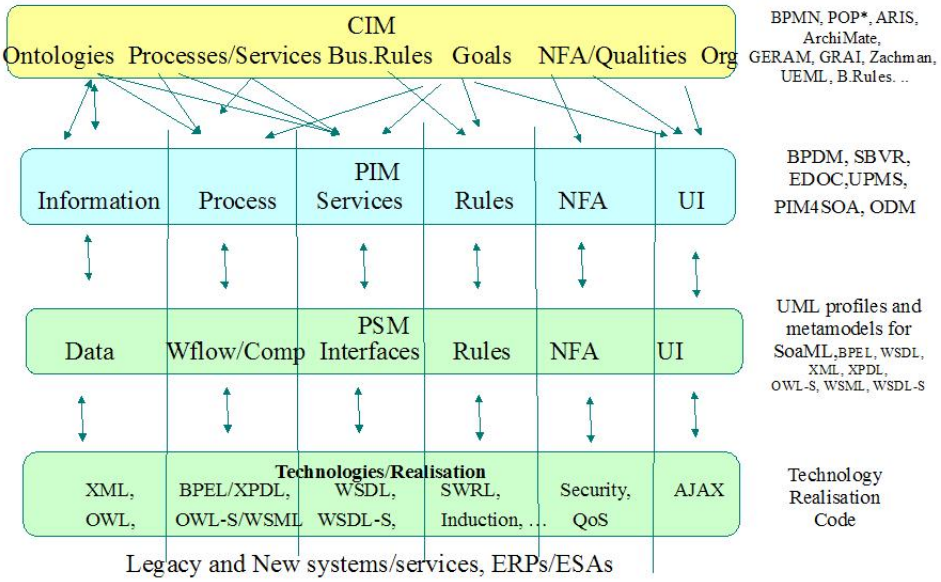


Figure 3: The reference framework

- Business rules -what are the conditions that govern the processes.
- Quality's/NFA -Non functional quality principles like security, reliability, performance, integrity and many more.
- Organization -The organizational units, roles, positions or resources involved in the business architecture.

5.2.1 Limiting Research area

There are a wide range of areas that covers most aspects of the enterprise architecture. However in this thesis I will focus mainly on the business services , goals , roles and business processes and not business rules , system management services, infrastructure and communication services, functions and non functional requirements.

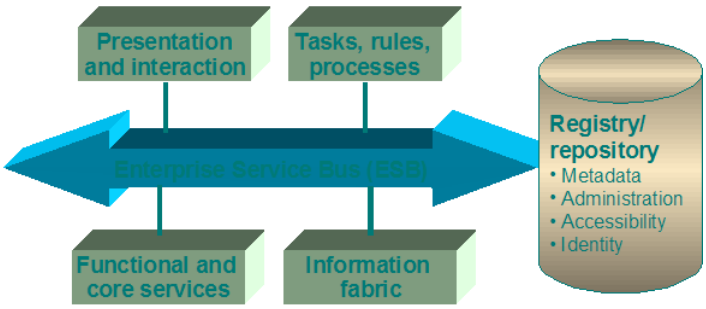


Figure 4: Tool

5.3 Tools

The tool support for methodologies is an important aspect of software development, also for SOA. Problems to look into in this area is how they address the ideas and the philosophy of their methodologies toolvise. Is it a closed or an open tool? By this I mean, is the tool service oriented or does it have a monolithic structure to keep the end to end process within the tool. See figure 4.

6 The case

The case used in this thesis is a typically Buyer Seller scenario, a service oriented project with the object is to automate the sales process of the EA. The organization have decided to launch a webshop based daughter company that buy and sell both physical and digital products. The shop tries to do business in the business to customer segment, and is using web technology to achieve this.

Part II

State of the art technologies

In this part the different state of the art technologies are described with respect to their process, tool, and notation and language. The description of the methodologies is a short version of the more thorough descriptions in the appendix. Additional information about the methodologies may be found there. The different methodologies where chosen because they are well known

and recognized in the research community. In addition they all have slightly different approaches to identify the services. By using all approaches and using them in an ordered sequence to target the various service perspectives, a process to identify the SOA will arise. Many different technologies considered as state of the art, where not used because of their similarity. Technologies that where not included in the analysis where Thomas Erl's SOA design patterns[10], ISE[11] from [12], SOMA [13] and SOAD[2] from IBM, Cummins Building agile enterprise[14]. Also methodologies considered analyzed were the SODA Service Oriented Development of Applications [15], SaE SOA Adoption and Excellence from [16] and [17], the Service oriented Migration and Reuse Techniques [18] and the Service Centric System Engineering [19].

7 ArchiMate

7.1 Introduction

Enterprise architecture is an important instrument to address this company-wide integration. It is a coherent whole of principles, methods and models that are used in the design and realization of the enterprises organizational structure, business processes, information systems, and IT infrastructure. A well defined Enterprise Architecture enables an organization to align business processes and IT operations with it's strategy. [20] An organization is a living and dynamic entity and the architecture must be flexible and able to respond quickly to changes and optimizational requirements.

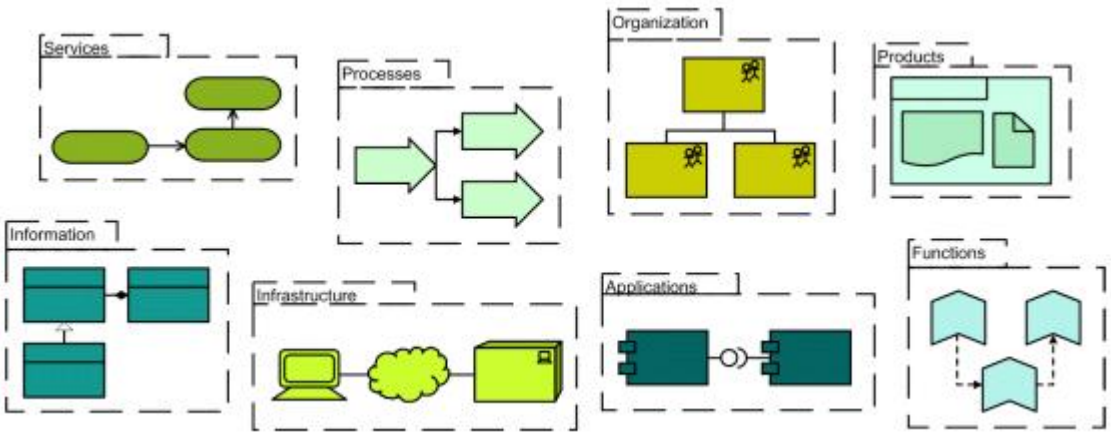


Figure 5: Archimate domains

Within companies various domain architectures can be found in Figure 5 , like organizations, business process, application, information, and technical architectures. Each of the various domains have been assigned their own set of concepts for modelling and visualization. The domains often overlap and use different notions to express the same ideas. Archimate is a tool and a modelling technique (language) that is trying to reach for an unified way of modelling enterprise architectures. All information and illustrations in this Archimate summary are collected from the book Enterprise architecture at work[21] and the ArchiMate[20] website.

7.2 Service layers

Service layers with services made available to other layers are interleaved with implementation layers that realize the services. There might also be internal services, e.g. , services of supporting applications that are used by the end-user applications. For specific layers more concrete concepts are used.

The architecture is due to this divided into three distinct layers:

- The business layer which confirms to the CIM level. The business layer offers products and services to external customers, which are realized in the organizations by business processes performed by business actors or roles.
- The Application layer supports the business layer with application services which are realized by (software) application components. This layer is equal to the PIM level.
- The Technology layer offers infrastructural services (e.g., processing, storage and communication services) needed to run applications, realized by computer and communication hardware and system software. This layer is equal to the PSM level.

The most important concepts of Archimate are shown above 6. You can clearly see the uniform approach across layers.

8 COMET-S

8.1 Introduction

COMET-S is promoting a model based methodology in a three layer architecture, Business model(CIM), Requirements model(PIM) and the Service architecture model layer. The starting point for the COMET-S methodology

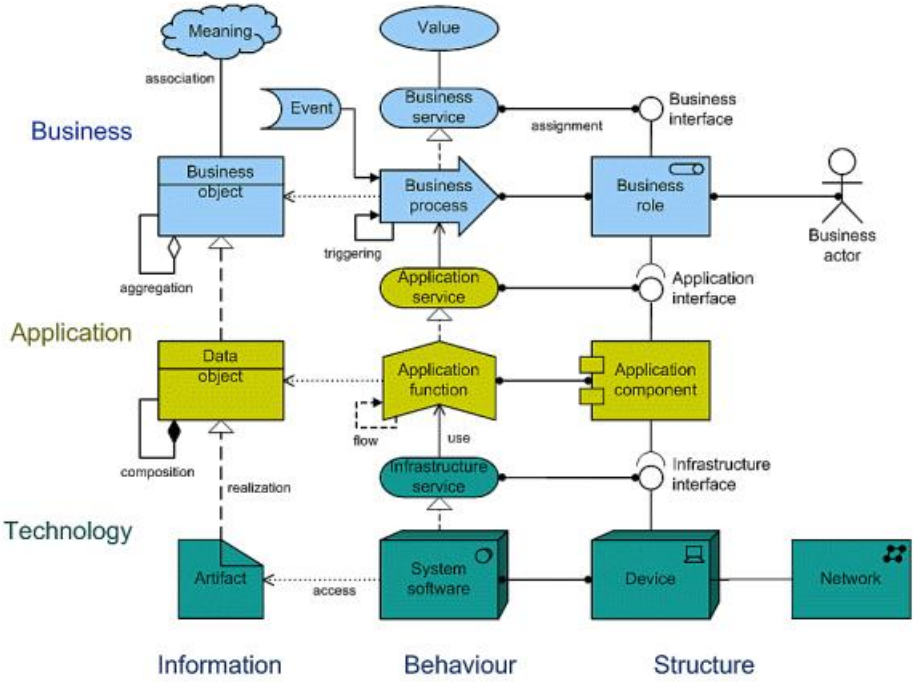


Figure 6: Archimate concept overview

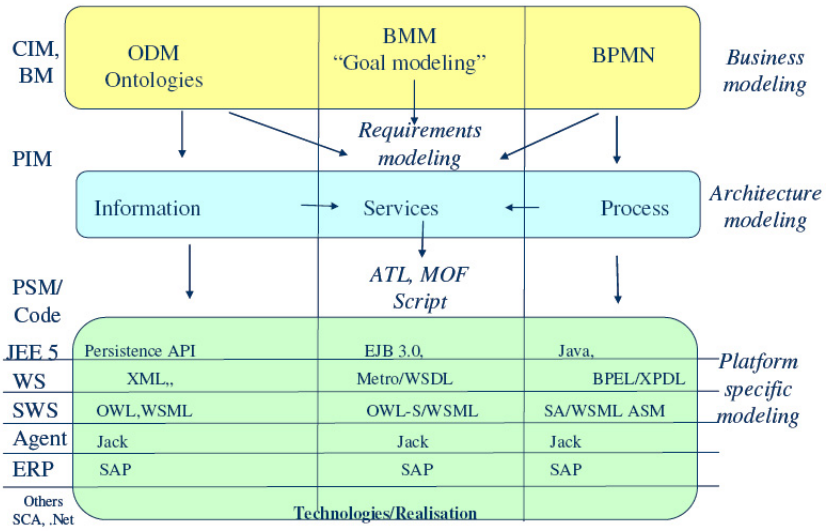


Figure 7: Modelling areas

(COMET for Services) is the existing COMET methodology as documented at the modelbased.net [22]. Comet-S provides some guidelines for the development process and notation of models. The processes and techniques introduced her is based on the COMET methodology developed mainly by several related projects ATHENA[23], SODIUM [24] and SWING [25]. The information about the COMET-S methodology is taken from the MDE for SOA[4]. The COMET-S methodology is compiled of four different modelling areas, business- model, Requirements model, Architecture model and platform specific model. It is using the newly available meta models from the OMG standardization projects. In particular for the CIM level BMM and BPMN are proposed, and for PIM the SoaML. The figure below gives an overview of the four main modelling areas of the COMET-S methodology (figure 7).

- Business model
- Requirements model
- Architecture model
- Platform specific model

8.2 Business model

The business modelling is used to outline and describe the part or role played by the product being developed. The business model consists of these work

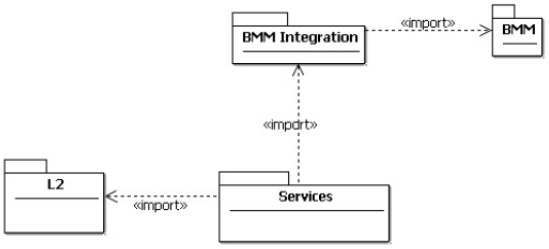


Figure 8: SOApro overview

products:

- Scoping statements including context statements, vision for change and risk analysis
- Goal model describing the business goals being realized through developing, implementing and using the product.
- Community model including business processes, role modelling and business resources.

8.3 Requirements model

The requirements model is identifying the system requirements including the functional requirements, non-functional requirements and constraints. The requirements model includes several sub models like the Use Case Model, a prototype, Non-Functional requirements and the BCE model. The Use Case Model consists of a System Boundary model and the Use Case Scenario model. The system Boundary model describes the System Boundaries, the actors and their responsibilities, and the services offered by the system.

8.4 Service Architecture Model

The Comet-S methodology is embracing the standards from OMG(Object Management Group), and are using the emerging UPMS standard as a framework for the service architecture. An open source implementation of the UPMS standard is in development by the European SHAPE IST project. Input to the standardization work has been provided also by earlier European projects.

The figure above 8 shows the overview of SOA-Pro meta model. The Services package is merged with the UML2 to extend the capability of service

modelling. There are several other Integration packages that extend Services with other OMG specifications like the BMM standard and others.

8.5 The PSM model

The Service architecture model is to be transformed into a platform specific model which contains the:

- Platform Profile Model which specifies the system in alignment to the actual technology profile for the specific platform.
- Component Implementation Model, which describes the implementation of the component specifications in a given programming language like JEE

9 SOA with OASIS

The SOA adoption blueprints can be seen as a set of functional descriptions of a service identification process. It provides a business problem statement, a set of business requirements and a normative set of functions to be fulfilled where vendor specific details are abstracted. It is supporting the use of the OASIS SOA reference model [6] and reference architecture[7], which spans over the whole Service oriented architecture with the intent to describe it's core information.

The Oasis blueprints consists of these elements
OASIS methodology
The OASIS methodology is highlighting the road to recognizing and describe which services needed to realize the business goals, objectives and its necessary capability's[5].
OASIS reference model
What is a reference model? A reference model is an abstract framework for understanding and describing important entities and their relationships, an ontology. The OASIS reference model has as a primary goal to create a foundation for a SOA vocabulary .It raises the question what is a "service oriented architecture" and try to address this. The reference architecture is more concrete and as a result it takes the concepts in the reference model and expand on them.
OASIS reference architecture
What is a reference architecture? A Reference architecture describes a domain with respect to its abstract achitectural elements from a non- vendor and technology independent view. As mentioned above the OASIS reference architecture takes the reference model a bit further, and also additional concepts are introduced due to the need for addressing the core questions of the Reference Architecture.

9.1 OASIS methodology

The methodology is in the context of the OASIS SOA reference model[6] and is addressing.

- Why services need to be defined.
- How to identify the shared and supporting services.
- The importance of a common language
- How to define interactions between services at a high level
- The categorization of services for management

The SOA methodology provides a business problem statement, a set of business requirements and a normative set of functions to be fulfilled where vendor specific details are abstracted. It is supporting the use of the OASIS SOA reference model and reference architecture, which spans over the whole Service oriented architecture. Well known architectural methodologies like Zachman[8] and others, begin the process of defining the business by investigating the context of the system or enterprise, the reason for it's existence and it's intentions. This is a good beginning for both the business strategy and architecture.

9.1.1 Big picture

One of the major goals of creating a service oriented architecture is to get the big picture. The big picture provides an overall guide to the enterprise, or a project, and will give foundation for splitting the capability's of the organizations or project into services. It will also give a deeper understanding of how change requests may be handled, and business change embraced through IT support. The methodology follow a four step process to develop an service architecture. The four processes are What, Who, Why and How. The methodology is mostly about the three first steps and only provides a direction for the fourth. The first phase What is about defining a scope of the services and what they should be. Number two Who externally is driving the services, and to whom do they interact. Third is the Why, why is internal and external services interact with each other. The forth and last is how, which is only given the direction for. How should they should be implemented.

9.1.2 Process

The blueprint gives some guidelines for how to execute the methodology framework for service discovery. I have above looked into the different elements of the blueprint like the different services and their decompositions, the support services and the common shared services.

Viewpoint Element	Viewpoint		
	Business via Services	*Realizing Service Oriented Architectures*	*Owning Service Oriented Architectures*
Main concepts	Captures what SOA means for people using it to conduct business.	Deals with the requirements for constructing a SOA.	Addresses issues involved in owning and managing a SOA.
Stakeholders	People (using SOA), Decision Makers, Enterprise Architects, Standards Architects and Analysts.	Standards Architects, Enterprise Architects, Business Analysts, Decision Makers, Standards Architects and Analysts.	Service Providers, Service Consumers, Decision Makers.
Concerns	Conduct business safely[8] and effectively.	Effective construction of SOA-based systems.	Processes for engaging in a SOA are effective, equitable, and assured.
Modeling Techniques	UML class diagrams	UML class and sequence diagrams, component and composite structure diagrams	UML class diagrams

Figure 9: The OASIS viewpoints

9.2 OASIS reference architecture

As mentioned above the OASIS reference architecture takes the reference model a bit further, and also additional concepts are introduced due to the need for addressing the core questions of the Reference Architecture. The OASIS reference architecture for SOA follows the guidelines in ANSI/IEEE std 1471-2000 recommended practice for architectural description for Software Intensive Systems.

The OASIS reference architecture aims to foster four principles. These are

- Technology Neutrality, platform independency.
- Parsimony, keeping it simple and minimizing the number of components and their relations.
- Separation of Concerns, loose coupling and stakeholder need to know basis.
- Applicability, to cover as many aspects of the SOA as possible.

The reference architecture for the SOA ecosystem provides tree main viewsseen in figure 76, first the **business via service view** that is the foundation for conducting business in the context of SOA. Secondly the Realizing Services view which addresses the detailed description of the participants, the services and its context. How is the Services realized at the platform independent modelling level. Thirdly, the owner view are addressing, evolving and maintaining a Service Oriented Architecture.

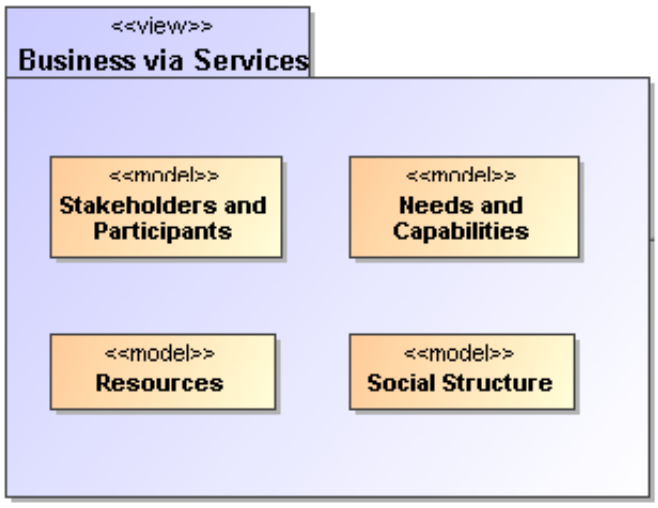


Figure 10: Business via services

9.2.1 Business via Services

About the Business via Service view, that that has a connection to the CIM level. This view contains four elements including models for their description, and are the Stakeholder and Participant model, resources model, Needs and capability model, Social structure model and its extensions.

Stakeholders:
People,decision makers, analysts and standard architects
Concerns:
Conduct business safely and effectively
Modeling methods:
UML-class diagrams

9.2.2 The Realizing SOA view

The realizing SOA view defines or describes the information needed to use, build, deploy, manage and manipulate a service. In addition to information and behavior models used to define the service interface. The description also includes information to decide if the service is fitted for service consumers needs. Information describing service reachability, service functionality,

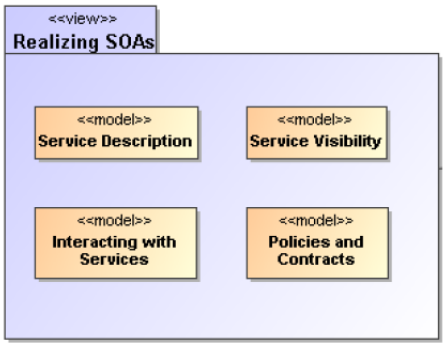


Figure 11: The Realizing SOA view

contracts and policies is also important model elements in this picture. Visibility and behavior models are also a part of this view.

Stakeholders: Enterprise architects, business analyst, standard architect and decision makers
Concerns: Effective construction of SOA-based systems
Modelling methods: UML class, sequence, component and composite structure diagram

9.2.3 Owning SOA view

Owning SOA view is about the different aspects of owning a SOA. A SOA based system is in a living and changing world and the environment the system is a part of is an "ecosystem" in a sense. To make the system adapt to the ecosystem, some management and governance are needed to ensure that all its components pull in the same direction.

Stakeholders: Decision makers,Service providers and Service consumers.
Concerns: Processes for engaging in a SOA are effective, fair and assured.
Modelling techniques:UML class diagrams

This view focuses on as seen in figure 72 three aspects of managing and governing SOA: security, governance and Services as managed entities.

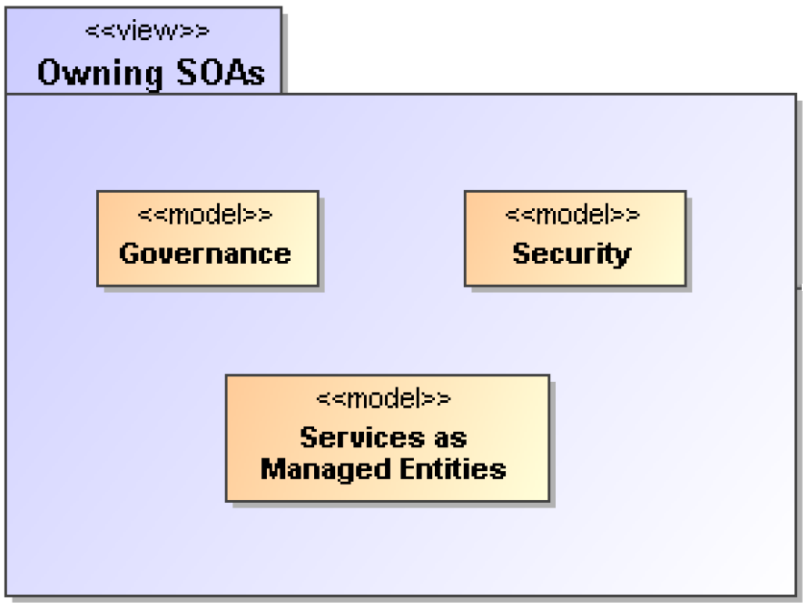


Figure 12: Owning SOA view

10 ARIS

ARIS is an acronym for ARchitecture of integrated Information Systems. ARIS is mainly a concept or a methodology that was developed by August-Wilhelm Scheer at the institute fur wirtschaftsinformatik at the Universitat des Saarlandes in Saarbrucken, Germany. It is also a computer based tool that enables you to model businesses on the ground of its methods. The methodology aims to close the gap between business theory and information and communication technology. This means expressing concepts of your business in such a precise way that it may be analyzed in detail and used as a baseline for the development of information systems. The ARIS tool represents the business by process models, services, systems, organizations, software, costs, data and so on. At last but equally important, is ARIS'es ability to model the relationships between them.

ARIS has several solutions for modelling and analyzing businesses. In addition to the standard Business architect/designer ARIS also has a SOA architect which integrates service description diagrams with the Business process diagrams. They do not take any position in the question of starting with services and building business processes around it, or to start with process and then integrate a service architecture on top. The important thing they argue is that there is a service architecture present integrated with the enterprise architecture.

10.1 Service descriptions

ARIS business oriented service description corresponds to the CIM(computer independent model) level, the top level services. This level contains no technical details and offers a top-down view of services. It describes what the service offers, not how it is implemented. The lower levels is a question about how it is implemented.

The first step in describing the services is the service architecture diagram. This first diagram is the breakdown of sales services.

10.2 Business Model

The business model is the way end-to-end processes is nested together for the purpose of achieving the businesses goals. In a business model one must be allowed to express the business logic of a specific firm. The business model should describe the different aspects of your business and it's relations. The ARIS tool or method if you like, defines these aspects as organizations, control, data, and its respective functions. In addition it should say something about the output products and services. The ARIS house is the representation of the concepts or views used to model the business model, which also reflected in the differing parts of the meta model.

10.3 Business Process

Business Process in ARIS context is the definitions of the tasks and the sequence of those task necessary to deliver a business function[26]. Business processes describe how a business pursues its objectives.

> "If it doesn't make 3 people angry it isn't a process[26]"

A process is either adding value to the business or fulfilling some necessary function. The process is a transformation witch takes input and generates the output as a service or a product. Central in ARIS is the process models flowcharts showing behavior, which is extended with enough information or semantics so that the process can be analyzed, simulated andor executed. This is the event driven process chain (EPC see figure 15) EPC and is the most used diagram, but ARIS also supports BPMN.

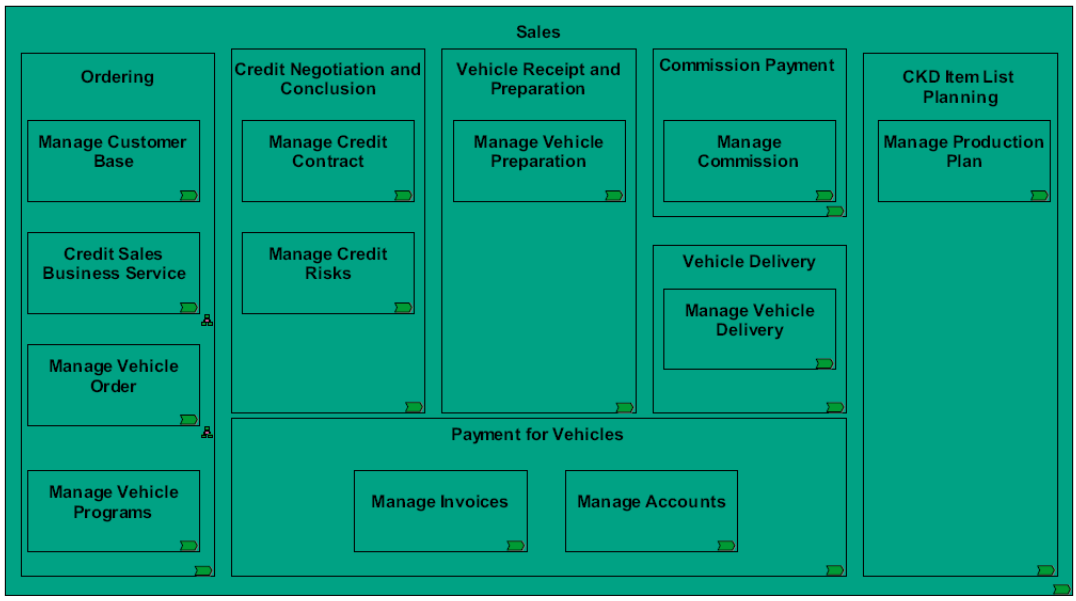


Figure 13: Service breakdown

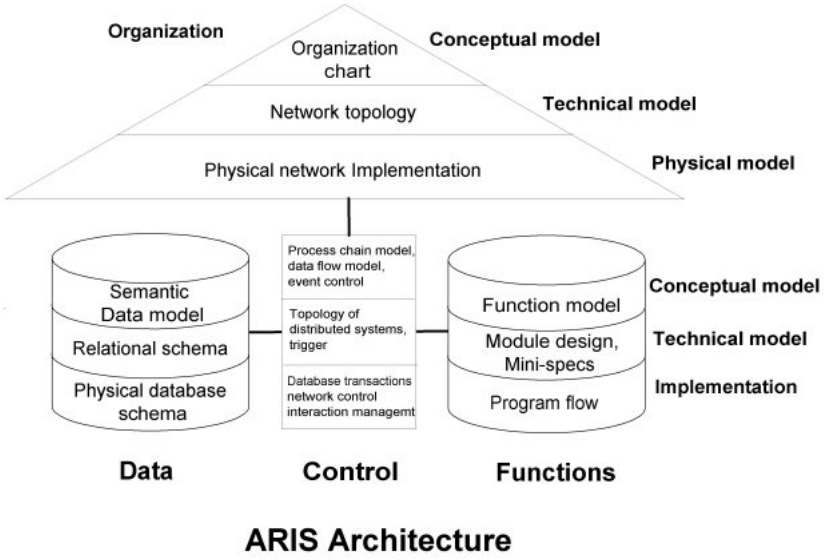


Figure 14: ARIS architecture

The ARIS tool also provides transformations from EPC or BPM, to BPEL or WSDL.

Part III

The Requirements

11 Introduction

11.1 Why these requirements?

In this part the requirements for analysing state of the art technologies and POSI are described. There are two kinds of requirements used. First the requirements REQ1.1-1.7 which are the different service identification tecniques which are the main focus of this paper. During the reasearch of state of the art technologies different service identification tecniques where identified. The techniques have different approaches and may inhibit abilities that represent a certain perspectives or types of motivation. Therefore methodologies that are using several techniques get paid off well in the analysis. The different tecniques are so to speak representing different motivations for the identification of services. Secondly the requirements REQ2-REQ9 that are defined by looking at what goals the state of the art

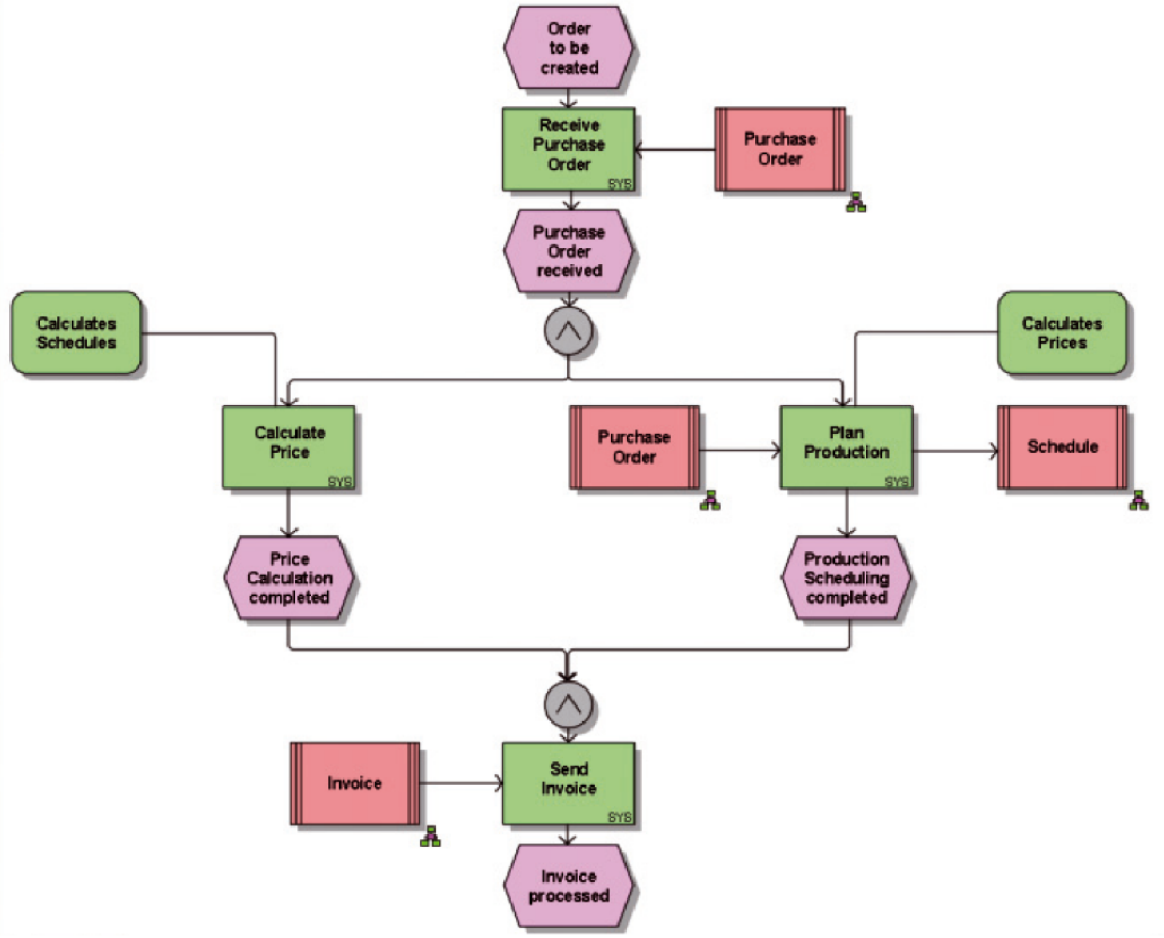


Figure 15: EPC diagram

technonlogies tries to achieve by developing the respective methodologies. Source for the more general requirements REQ2-REQ9 are mainly inspired from the requirements of ongoing OMG work with SoaML[27].

11.2 How are the requirements applied?

The list of requirements will be applied to the methodologies and analyzed to give a meaningful classification and to highlight the areas of a methodology that is worth bringing into POSI. Later in this document they will also be used to evaluate POSI. The concepts most important to create a Service architecture model with focus on the participants needs, goals, capability's and values are attempted represented in this requirements list. When looking into how services are identified and how they evolve in the Enterprise architecture one must decide what information is needed to describe the services both on CIM and PIM level. The information that we must retrieve is the major stakeholders and their goals, needs and capability's or some other motivation. In short the reason for their existence. It is important to align the services with the business philosophy and goals. This information can among with other techniques be modelled as a hierarchy of needs and capabilitys, that would give a base for identifying the services needed, their responsible roles and their relationships with other services and participants.

12 Areas to address

The requirement must address the following issues that is central for the identification of the service architecture.

- The services identified must be grounded by the business strategy or some motivation model that gives resoning for the exitence of a service and it's attributes.
- The language and notation should be aligned to the comprehension of the persons who designs, or is responsible for designing the different aspects of the Service architecture.
- Enabling composing services from other services
- The ability to extract service identity, services provider and consumer and service naming conventions.
- The Service specification that should specify the services independtly of how they are provided or implemented.
- The Service contract which describes the contract that must be met by the service specification or realization.

- meaningful lossless transformations between CIM and PIM level concepts
- Specification of high level service data that are used internally in services and externally data structures comunicated between provider and consumer.
- The methodology should address how to derive the Services. Service identification can be done in many different ways.
- Addressing reuse of existing services and artifacts is an important requirement.

13 The lists of requirements

13.1 REQ1: Service identifcation techniques

Using Service identifcation techniques is a way of ensuring that the service architecture are aligned to the stakeholders plan for their business. This means that the methodology's should address how to conduct the identification of what services are the ones that together will solve the challenges of the business or organization. This is the main requirement and the focus of the analysis. The services should also be grounded in some sort of a business motivation model, that contributes to bridge the gap between the business and technology stakeholders. In the literature covering the different approaches to the creation of service architectures described in the part of state of the art technologies, there are several ways of conducting the service architectures. The most popular and most noted ways of finding the services of a service oriented architecture are the process driven, use case driven, through analysis of collaborations, capability-need or product-value driven and a few others.

13.1.1 1.Process driven

Using the notion of end to end processes or business processes that is compiled to offer a portifolio of internal and external services. Processes and services represent different aspects of the same system. The first is a view of how and why participants interact to use and consume products and services. The processes represent a view of what activities or tasks participants are performing to provide those services. This is an approach that is seen in the ARIS platform, where system functions are seen as services that realizes business processes.

13.1.2 2.Capability, Need, Goals, Product and Value

The different approaches in this family of service identification techniques, are using the notion of individual functions organized in a service hierarchy. The

service hierarchy represents the top down service view, showing the most important services, the intendent use, dependencies and their relations. First there is the notion of needs that are covered by one or more capability's. Secondly is the notion of a product compiled by different services, where the product is representing a value that the service consumer is interested in. Thirdly is the goal based approach where the service architecture is defined by a hierarchy of loosely coupled goal model. Ends are the desired outcome e.g. goals and objectives that the business are trying to achieve. The steps taken to achieve the goals and objectives are the means, tactics, strategy, business policies and business rules. They are structured ways of expressing what a participant want and offer.

13.1.3 3.Collaboration driven

This group of service identification techniques are taking advantage of the fact that a service architecture is realized by two or more participants are working together and communicating through specified channels or ports. There are a few different distinct techniques in this group. First using the notion of communities having interacting collaborating participants where their relationship are regulated through service contracts. The service contracts are more or less binding and are specifying how they are to interact in order to fulfill some common purpose. Secondly there is Fred Cummins[14] Business unit concepts that is a hybrid of the collaboration and the capability based service identification techniques. Having roles grouped with respect to their capabilitys. The capabilitys that are similar and operates in the same context belong in the same service. Similar capability's operating in different context are in the same Service unit. A service unit are the container for the services.

13.1.4 4.Use case driven

The use case driven approach has arised from the object oriented analysis and design community, and is proven useful when collecting the functional requirements in a software development process. Use case models with their associated UML-diagrams are outlining key functionality in a system , and the participants or actors in use case context.

13.1.5 5.Business Service driven

The business service driven approach are used to express the top level services important to achieve customer satisfaction and internally processes. This widely applied approach is used by ISE, a Zachman inspired methodology sprung out of a German founded reasearch project Theseus[12]. The service concepts are seen upon as points where the business are exposing its internal and external interaction between services providers and the consumer.

13.1.6 6.Service Classification driven

The service classification considers the perspectives of an architecture and creates services that realizes each aspect like interaction services, process services, information services, rule services processing services, infrastructure services and management services.

13.1.7 7.Bottom up driven

Bottom up driven is a strategy to identify existing services that matches the requirements of the ones identified in the ongoing process. This process should be supported by having the notion of service categories. This because existing services are much easier to identify when you have categories that can classify the service.

13.2 REQ2: Ability to extract core service attributes

The ability to extract core service attributes is finding the service name, service provider and consumer and is high level descriptions of the services. This description is describing the services in a top down perspective. Can the attributes be derived from the motivation model or must the developers manually create attributes and relations. The core attributes is: giving the service a reasonable humanly name and describing its associations and relations with other participants and other services. In addition also the messages sent between them.

13.3 REQ3: Modelling language and notation alignment

to the designers expected needs and capability's. It is important that those who are designing the different aspects of the Service architecture are able to understand the related tool, language and notation of the design process. If else they will not be able to express or document their knowledge about the domain.

13.4 REQ4: The Information model

The information model is mainly the representation of the resources, and the description of messages that are sent between the services.

13.5 REQ5: Service specification

includes a more indepth view of the service. The modelling language must be able to express possible many Service interfaces, with its respective service

operations that are all available across distributed, concurrent systems. A service specification should also include the operations pre and post conditions, parameters and exceptions. In addition the Service interaction points through which service interfaces are provided and required, in order to distinctly identify consumers and providers. Last the service specification should include owned behavior. The service specification should also include most of what is considered as the service contract. The **Service contract** includes behavioral information about how interaction is to be conducted according to the services respective roles. In addition constraints or objectives to be meet, and descriptions of participants and their roles. Also included are behavioral rules and the interactions between the roles. This requirement is partly out of scope, and will not all be explored in detail.

13.6 REQ4: The Information model

The information model is the representation of the resources, and the messages that are sent between the services. The model must define the main domain concepts and their attributes.

13.7 REQ6: CIM to PIM transformations

Is the CIM to PIM transformations easy to visualize or is there a poor coherence between source and target metamodels, or is the PIM model only an extension of the CIM model?

13.8 REQ7: Process support

Is there described a path from start to end considering phases, roles and disciplines of the development process.

13.9 REQ8: Tool support

Is tool support existing for the methodology. Tool support is having artifacts that help you to presist, implement, execute and share artifacts, models and other documents related to the development process.

13.10 REQ9: Holistic View

Does the methodology provide a holistic view of the system, or are the models adressing only parts of a system. Does it support several ways of finding services. Does it connect different parts of the architecture as one universe or are the different aspects unreferenced.

The possible score on these requirements is from low 0 to high 2. 0 is given when there exists a formal document of the services but no model. 1 is given when there exists a formal model of the services, but where the ability to reference the service architecture to other aspects like organization and business rules and policies are not there. To score a 2 the methodology must provide a compatible language and notation to link inn other aspects. The ability to model both synchronous and asynchronous messaging should also be provided.

Part IV
The Analysis of COMET-S

In this part the COMET-S methodology is analyzed against the proposed requirements from part 13. The analysis is done by evaluating the state of the art technologies described in part 2 with the requirements described in part 3.

14 REQ1: Service identification techniques

When taking the COMET-S methodology under investigation we can see that there is a rich variety in the language and notation. Specially at the CIM level the methodology supports many ways of expressing services and their relations to other aspects of a system. When looking at the first CIM requirement, the language and notation for describing the business goals are the language provided by OMG's BMM(business motivation model). The main advantages of using OMG's metamodels beside that it inherit a rich set of concepts to model aspects of the business plan, are that it is more or less methodology neutral and that it contains role references to OMG's organizational, business process and business rule models. Key concepts that are central to services and their use are the ends and means for the business. Ends are the desired outcome e.g. goals and objectives that the business are trying to achieve. The steps taken to achieve the goals and objectives are the means, tactics, strategy, business policies and business rules. The strategy and tactics are concepts that manifestations of what the business are planing to do. To become the marked leader in for instance online sales of CD's is he vision, if revenue is to rise with 20 percentage, which is the goal. The objectives is then to give the customer access to searchable products, to reserve products, pay products and enable delivery and schedule production if necessary. The associated courses of action is mapped as services. To achieve these ends it is necessary to have a course of action that is to sell cd's online. Selling cd includes several other services, products must be searchable, login functionality, shopping cart functionality, payment and delivery services. See figure 16

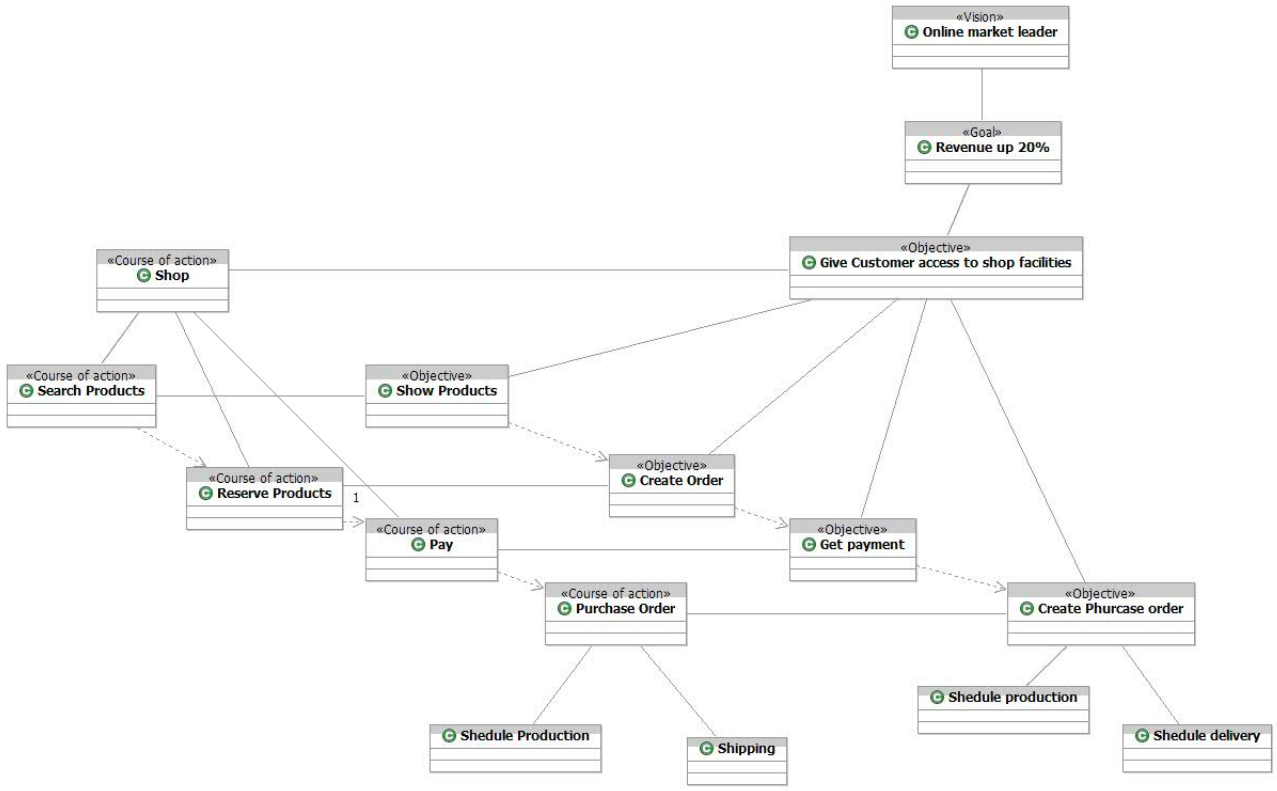


Figure 16: BMM example

All these are means to reach the desired ends and can be refined and modelled as services. The COMET-S methodology is not only providing a language and a notation to express the business service architecture, but also a suggestion of how to extract the service components. The Comet-S methodology is using the BMM language to model the part of the Business model that details the business processes and business resources that are relevant to the Product. The BMM goal model describes a hierarchy of the business goals within a particular area of concern. Beginning with the goals of Business Stakeholders in developing or buying the Product, and all the way down to the concrete business goals of the stakeholders. The goals that are identified within the "Community" are classified as either a resource service or the enabling behavior's Business services that support the goals defined in the Goal model. As you see from the figure 17there is the concept Means that has a sub concept "Course of action" which is the solution to achieve Ends, the desired result.

The course of action concept can be modelled to represent a hierarchy of services. To achieve each desired result is achieved by doing one or more cources of action. The strategy could be mapped to services and tactics to the realizing processes, all connected to a goal or objective. The COMET-S methodology also supports the usage of Use case diagrams and collaborations for modelling services. As you see in the figure 18 there is a simple service architecture derived from the BMM concepts.

The methodology also talks about classifying services with respect to some aspect. Information services are created from entity's in the resources model. Business services are created from the <manage> stereotyped use case. The same notion can be used for goal model and collaboration diagrams. The COMET-S supports as mentioned the collaboration concepts and it's UML-diagrams which is not expanded here. The same services in a BPMN diagram in figure 19. So the evaluation of COMET-S's service identification techniques gives a 2 for Use case, 2 for collaboration, 2 for goal driven, and 1 for classifying services because its kind of uncomplete.

15 REQ2: Core Service attributes

The ability to extract and model services and their core components are the second requirement I choose to investigate. This development process are

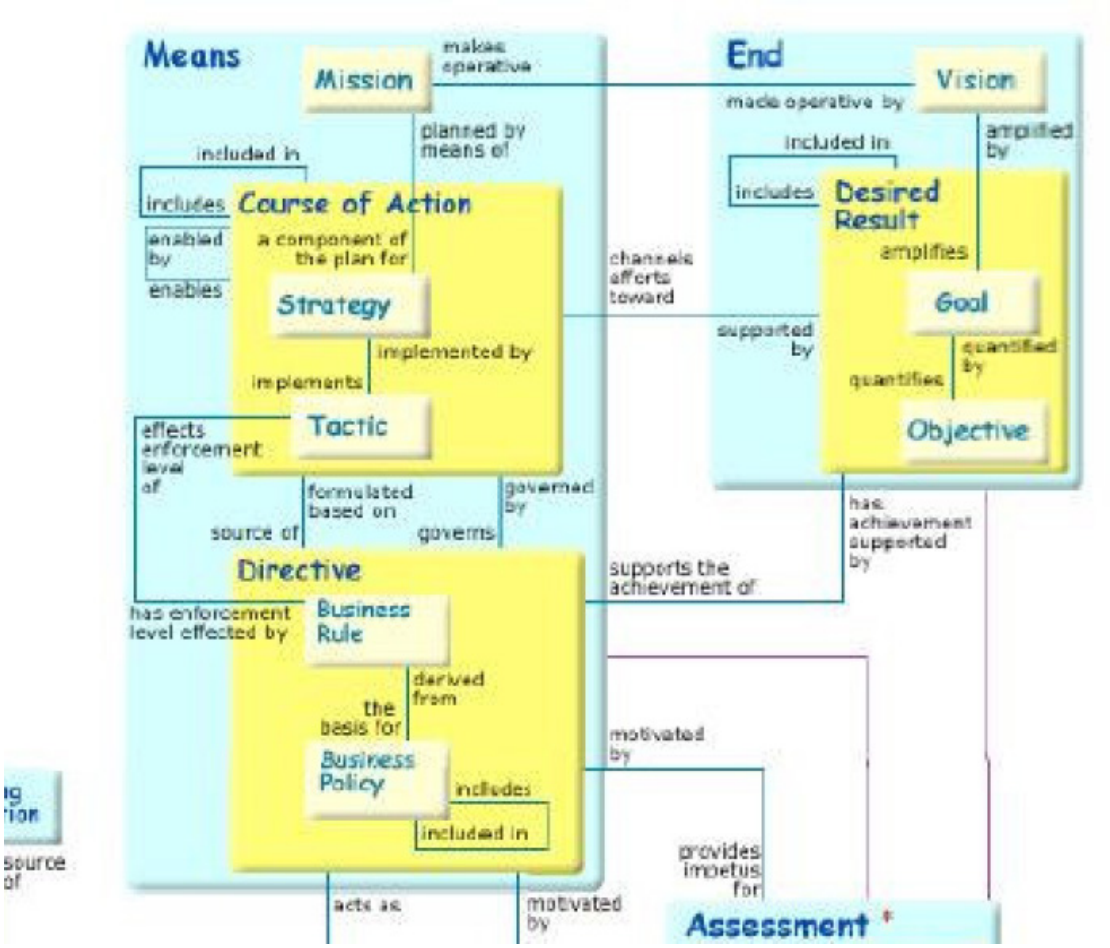


Figure 17: BMM metamodel

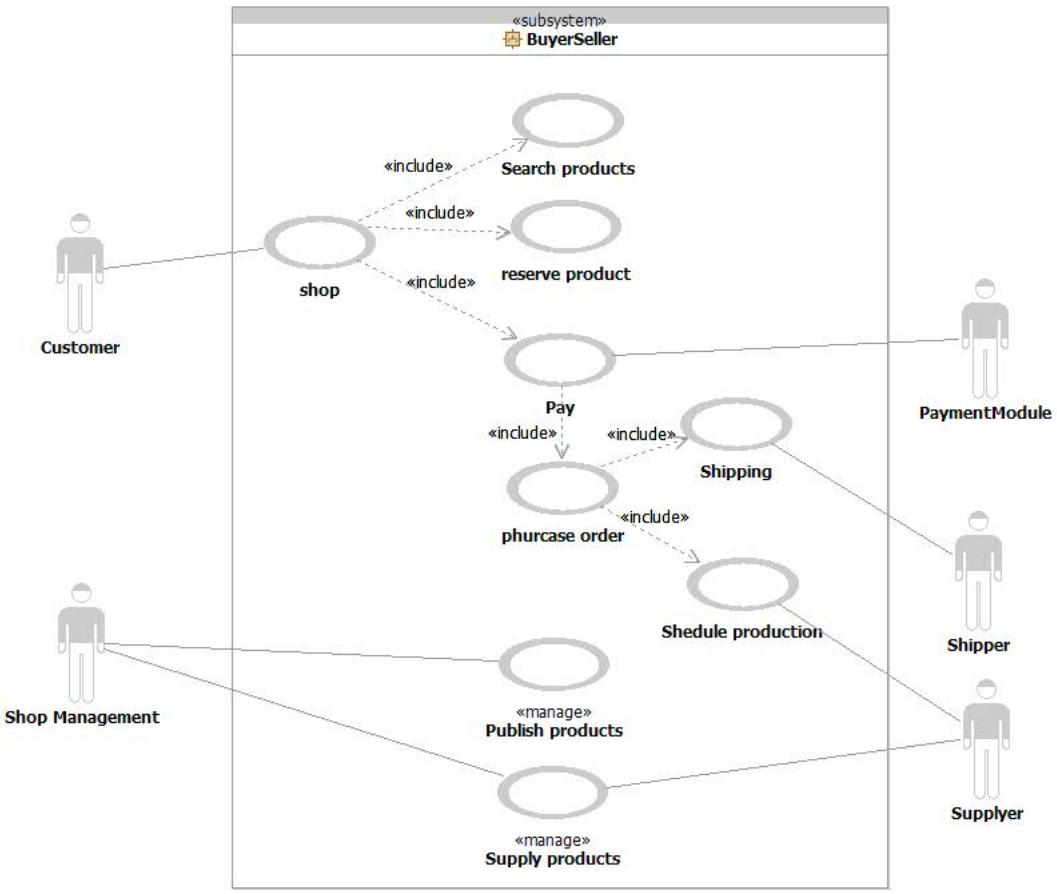


Figure 18: Use case Service example

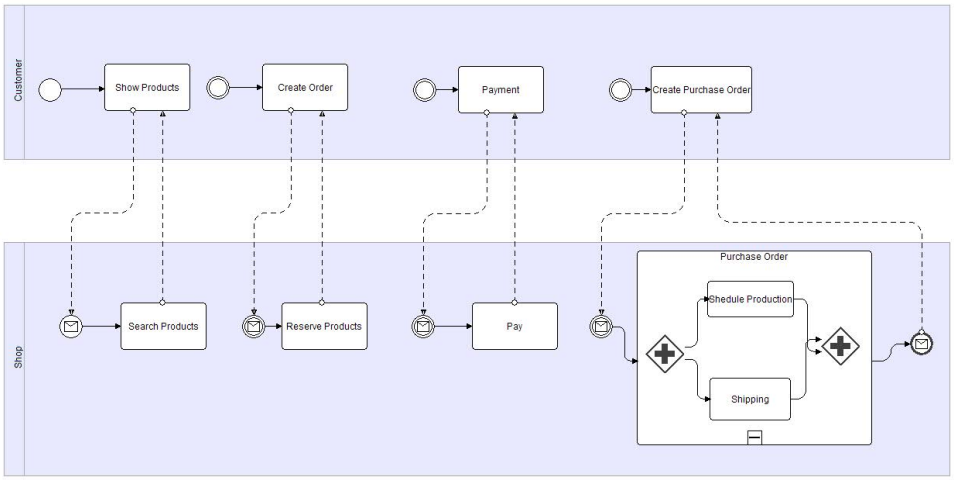


Figure 19: BPMN example

supported by the BPMN process language that are describing the services and it's realizing processes in several levels of abstraction. Used to model the organization and it's roles the OMG Initially in the COMET-S methodology this is done by a brainstorming process or any other creative process a wide range of stakeholders can participate in. The roles involved are expressed by an organizational structure metamodel. The COMET-S methodology talks about creating the process model by identifying behavior that are needed to achieve each of the desired goals. These behavior may be structured in categories or hierarchies addressing one or more products and may be arranged as services. The BPMN language are a really flexible language, and the 2.0 version and higher comes with the ability to model collaborations in addition to the traditional process model mapping pools, swimlanes tasks and messages. 2 points.

16 REQ3: Notation and language

How are the notation and language aligned to the ones who are conducting the work on different aspects? Well, when looking at the Goal and motivation model the main stakeholders are business owners or business analysts, consumers and a system analyst. This kind of documentation is likely to exist in organizations of some size, and are usually existing in the form of a word document or a power point presentation. So the "sketch book " capturing and formalizing this information must fit non technical staff needs and their level of comprehension. One problem is the different notation possibility's in BPMN which comes clear if you consider figure 19 and figure 20. The **BMM** model

has concepts that is easily understood by business people. There may be a need for some participants from the business side to learn some basic modelling skills, e.g. how concepts connected with associations form a graph that express a picture of a realworld problem. When it comes to developers and system analysts that have experience with modelling languages. The issue for them would most likely be to understand how the concepts of how the business is conducted, and the vocabulary used to describe the domain. Not the ability to understand modelling. The **BPMN** language and notation are quite intuitive, and are well aligned with it's needs. This is a rich language for process and service modelling and there were few things I thought were missing. The processes are modelled as a flow of tasks in pools and swimlanes with the possibility to branch the graph with gateway operators like "exclusive", "inclusive" and "parallel". The flow however cannot go cross pools which is quite intuitive. One must pass messages over lanes. The messages are either collected through a message event or directly to a process. To denote the messages between different elements it is possible to attach a data object to the message connection. The processes can be nested through the notion of a subprocess like in figure 19. To represent a state that denotes the result of a task the event is present. However one of the things that one could feel is missing, is the notion of and the distinction human or system tasks. This could ease the transformation of the interfaces between human and computer, and computer versus computer. But this again depends on how the pools and swimlanes are used. Are they indented to represent participants, services or both? This makes BPMN flexible and more ambiguous. In the diagram 20 the pools are representing participants and the lanes are services. Each pass of message represent an operation or a method. In the evaluation further on, this is assumed as the case.

The **SoaML** metamodel is well fitted to hold the information about the service architecture, and contains enough concepts to represent the core attributes at PIM level. Talking about the concept of participant, provided services and required services. Taking these two CIM level modelling languages in COMET-S and the SoaML in consideration, the conclusion is that they are well fitted to be used to carve out the business architecture, with only some minor shortage. This is a subjective evaluation but gives a pointer in a certain direction. The methodology gets the value 2, because the language and notation are well aligned.

17 REQ4: Information model

The information model in COMET-S is the resource model which in CIM level is a collection of main things or concepts of the domain that are relevant to the service architecture. The information model in COMET-S is an UML class

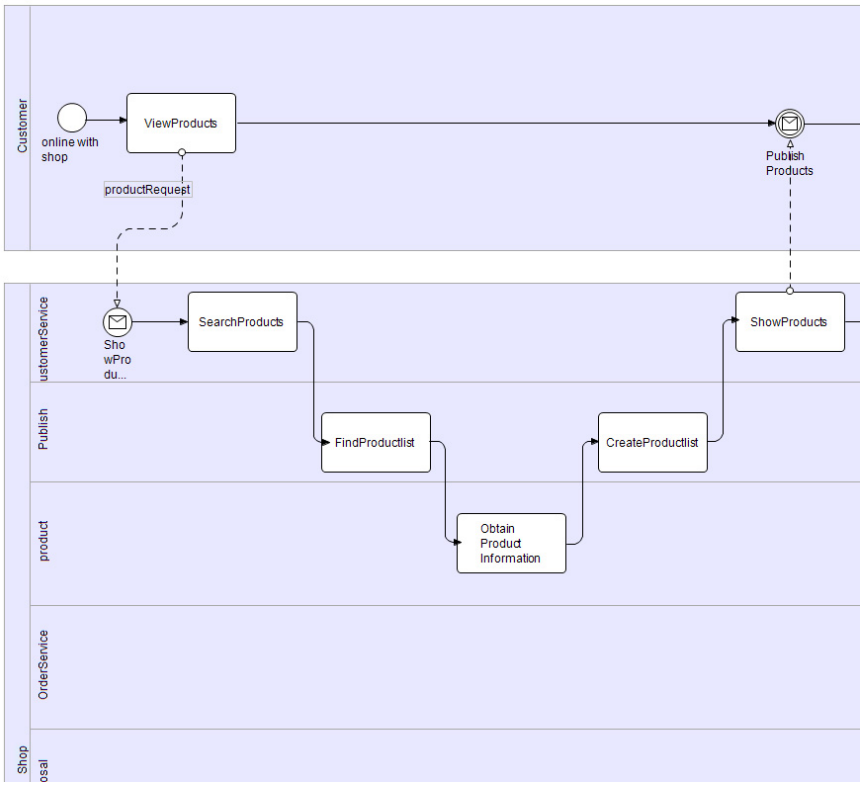


Figure 20: Example of a BPMN service description

diagram and can be linked to an ontology model expressed in OWL[28] or with an OWL UML profile. It could also be represented as a native OWL model. The information model and ontology are representing the vocabulary that is used to talk about the domain. Specially the participants, resources used, and messages sent including their arguments, should become entities in the information model. Those things considered this is a good representation of the information model that is proposed in COMET-S, and this gives the methodology top score 2 on this requirement.

18 REQ5: Service specification

Does the information captured in the process of designing and documenting the service architecture contain enough information to transform the business models to platform independent models. To generate the multiple interfaces with it's service operations are done by looking at the messages going in and out of a service lane. Some of the pre and post conditions can be derived by

the sequence of tasks or services or events. Services located before a certain service in the flow are pre conditions, if it is not an optional operation or service. Parameters can be expressed as dataobjects attached to the message flow. Exceptions have it's own symbol in BPMN, and is included directly in the diagram. The behavioral parts of service contracts are also derivable if one follow the messages and flows in the BPMN diagram focusing on the service lanes, in addition to how and who the service are communicating with. There is a good correlation between capabilitys and needs, which gives a score of a good 1 almost 2.

19 REQ6: CIM2PIM transformation

COMET-S are specifying the ATL as the model to model language, and MOF script as model to text language. These are both well known and well tested languages and will satisfy the need for transformations. The PIM model that COMET-S uses for describing the service architecture is the service oriented modelling language(SoaML). This is an UML profile and a metamodel describing a framework for describing Service Architectures at PIM level. The transformations from CIM to PIM are quite obvious. The Participants in CIM diagrams are mapped to participants in the PIM metamodel. The services are either mapped a class of Service which is the participants provided services, or it is a Request that is the required services. The message and flow connectors between the pools and lanes are giving us the interfaces. Even thus the mappings are quite obvious between the BPMN diagrams and SOAML there is not an exact procedure described to derive the service architecture from a BMM model. This would require that the service hierarchy must be present in the BMM diagram, and this means that one must model the BMM diagram with this in mind. This gives the methodology a 1 point score for this requirement because it is has some kind of solution, but not a complete path from A to Z.

20 REQ7: Process support

How is the language and notation supported with development roles, discipline and phases. The methodology is supported with modelling objectives, and methods and techniques for each delivery in the development process. It also proposes guidelines for who the major stakeholders are, in relation to the methods and techniques. The process are described thoroughly, and is only lacking a precise description of how to derive the right Business services from BMM model. The score is therefore 1.

21 REQ8: Tool support

There is no dedicated tool that support the COMET-S methodology. There is developed some tool support for BPMN diagrams in SOA tools package for eclipse, and there are also developed some other standalone modelling tools for BPMN and BMM. This gives the methodology a 1 in the evaluation because the tool support are existing but not as an integrated holistic methodology.

22 REQ9: Holistic View

. Does the methodology has a holistic view of the system or is the models addressing only parts of a system. Does it support several ways of finding services. Does it connect different parts of the architecture as one universe or are the different aspects unreferenced. The COMET-S methodology is viewing the Service Architecture in a holistic view, joining the business model to the platform independent model and further to the platform Specific Model. It advocates a multi perspective view on architecture, separating concerns but still keeping it holistic. The methodology scores only a high 1 on this requirement because many of the transformations are not completely defined.

23 COMET-S summary table

This table sumarizes the points for the COMET-S methodology.

Part V
The Analysis of ARIS

In this part the ARIS methodology is analyzed against the proposed requirements from part 13. This section will try to outline the different parts of the ARIS methodology.

24 REQ1: Service identification techniques

With ARIS Strategy Platform, balanced scorecard systems and use of key performance indicators, can be established and business processes aligned accordingly. This is a really complex and thorough process and almost requires an economic background. The strategy platform does not directly

Requirement	Key elements	Value
Service identification techniques	Bottom up Process Capability Calloboration Use case Business Service Architectual aspect	0 0 2 2 2 0 1
Service attributes	BPMN and BMM	1
Notation and language	UML activity diagram, BPMN and SOA-Pro	2
Information model	UML-class diagram and OWL	2
Service description	UML activity diagram, BPMN and SOA-Pro	1
Transformations	Partly described	1
Process	Comet-s phases, roles and disciplines	1
Tool support	Some eclipse tools	1
Holistic view	Aspects, layers, tools, process	1

Table 1: COMET-S summary table

support service identification, but are used as a guidance for the creation of business process hierarchy in what ARIS is calling Service oriented business process modelling. The services are created to support the business processes and indirectly to support business motivation. This is classified as process driven service identification. One of ARIS greatest strength is it's ability to focus on and model processes. 2 points for ARIS process driven service identification. ARIS also describes a bottom up approach to identify matches between the services and existing service architecture. Bottom up 2 points.

25 REQ2: Core service attributes

In section A it is said that ARIS does not favorize either starting with services or processes. However in documentation the top down bottom up approach is the method described. In figure 21 the business processes are broken down in the left hand pyramid to level 3, where the business functions can be mapped down to the appropriate service. The transformation of EPC diagrams start at the lowest level to a BPEL process. After this is done the

BPEL processes at level two in the right hand pyramid can be combined as more comprehensive business services. All processes are combined in a higher level BPEL process at level 3 in the right hand pyramid.

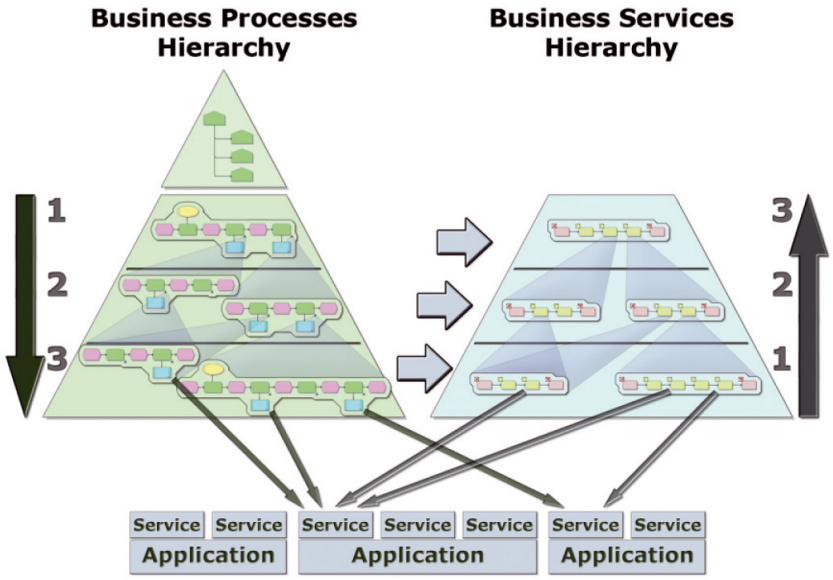


Figure 21: Top Down and Bottom up

Like mentioned above ARIS advocates the creation of high level business process models, and using them to reveal technical properties that technical services need to have in order to [3] support defined business activitys. The diagrams used in ARIS does support this requirement very well. ARIS supports not only EPC, but also BPMN and both of these languages are rich enough to hold the information needed to extract core service attributes. VAC diagrams are also used to model high level service or process maps.

The BPMN language are discussed in the previous section IV. In addition ARIS platform now supports ArchiMate which is discussed in VII. As we see in figure 22 which is a simple purchase order process. The process model includes the "plan production" step. The scheduling object represents the service that will automate the process step Plan Production represented by a function module. The function's input data is the purchase order data object and output is the schedule data object. The service can be detailed further in associated UML-diagrams. The participants and roles can be connected to the

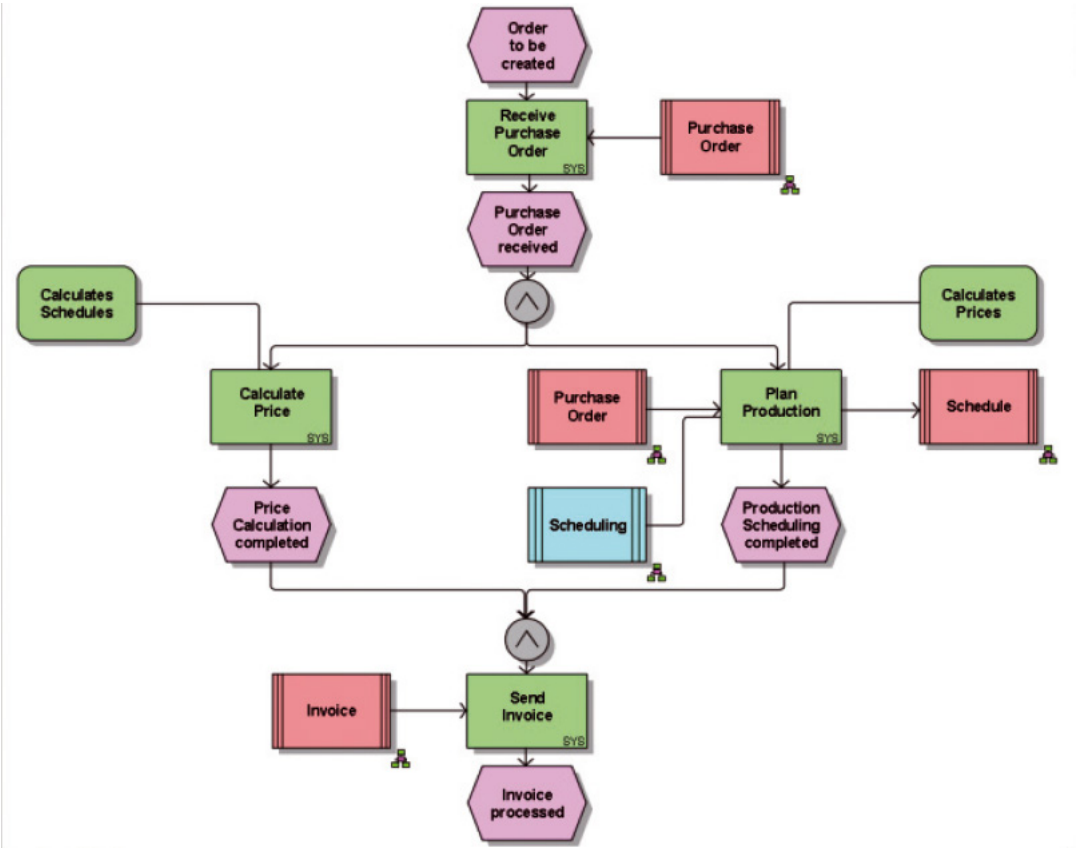


Figure 22: Example of a SOA Process modelling with Service

function module in the EPC diagram or be connected as seen in the service assignment diagram 39. This is a really good way of organizing the organization. The process and service model is in center binding the information model, organization model and functional the model together. However the BSC is not giving the services directly, but indirectly through processes. 1 points.

26 REQ3: Notation and language

The notation is intuitive and simple. The different models are woven together with the EPC diagram that is both easy to use, and it provides high flexibility by making it easy to reference and nest different models together. The symbols used are simple and this leads to little “noise” in the notation. However the 2 points.

27 REQ4: Information model

The information model is either imported XSD files, modelled as ERM or uml class diagram. As seen in the diagram 23

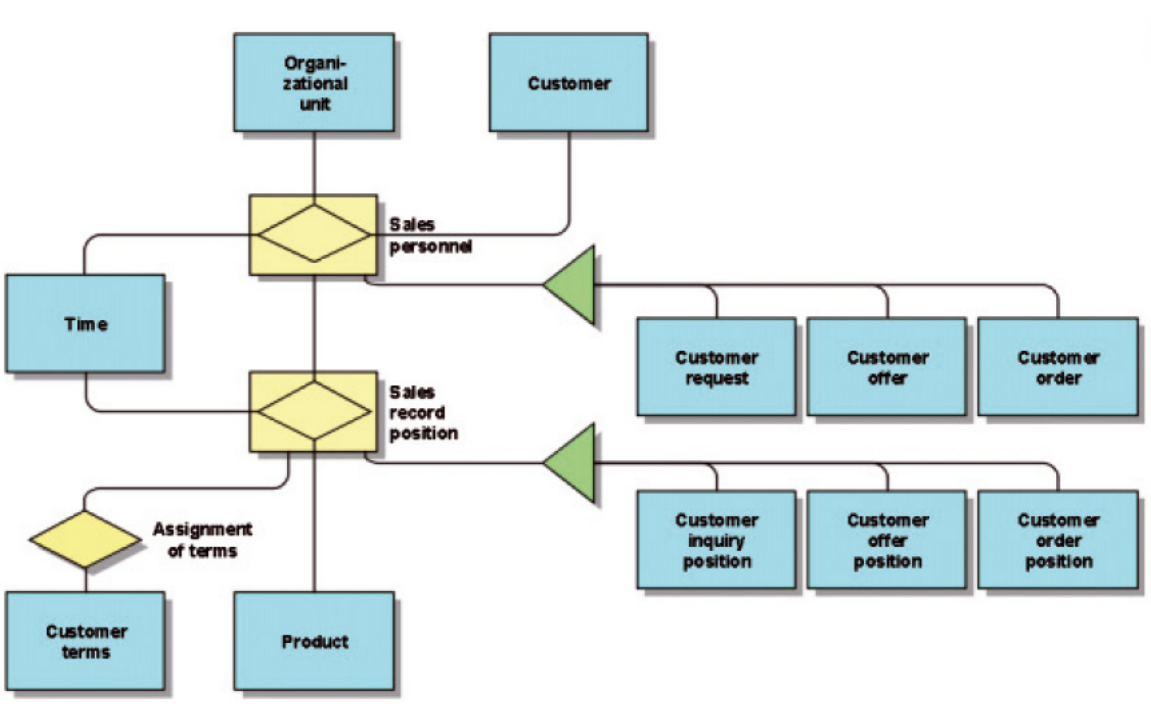


Figure 23: Example of a ARIS data model

It shows a description of the data model for sales. This data objects is reused in the process and service descriptions. Typically it is reused in process descriptions, describing input/output data related to tasks or activities, data flow, and information exchange between the participants and applications. The downer is that it is so many different diagrams that you must have some pre knowledge of the methods used, and which diagram can be inter referenced. 2 points.

28 REQ5: Service specification

As mentioned earlier the process and service models are nestable and hence it is possible to construct models on different levels. This enables the possibility

of denoting the behavior of the elements involved, gateways and events make pre and post conditions. The message flows and access diagrams with in and out data objects, services, processes and roles give the service interfaces. This covers this requirement pretty good, 2 points.

29 REQ6: CIM2PIM transformation

ARIS is using standard javascripts for model to model transformations, giving access to all java classes. There is a simple script editor in ARIS with syntax highlighting, auto-completion, Integrated debugger, various step control options and macros. There is some prewritten transformation scripts in ARIS, like from EPC to BPEL or BPMN to BPEL. It is also a possibility to use XSLT for transformations. 2 Points.

30 REQ7: Process support

There are some documentation on several ways of conducting the development process in ARIS, but there are also the inherent phases that the tool structure represents. The ARIS architecture consists of the design platform, implementation platform and the controlling platform which represent the whole development lifecycle:

- Strategy platform that is used for defining business strategies, implementing them in operational processes, and performing continuous monitoring of target systems.
- ARIS Design Platform for distributed modelling, simulation, optimization, and publishing of business processes and managing IT architectures.
- ARIS Controlling Platform for dynamic monitoring of current business processes, implementing corporate performance management systems, and establishing an enterprise-wide compliance management system.

When looking at roles and disciplines there is not much documentation covering these areas. 1 point.

31 REQ8: Tool support

The ARIS modelling tool support is really grasping over a wide range. The ontology could be modelled in a Technical term diagram, but also the repository of all objects created are in a wide sense also a part of the Ontology. The goals aspect is supported by SCORE-card analysis, and KPI diagrams are

among diagrams used to describe them. For modelling the top level business processes, ARIS provide the Value added chain(VAC) diagram and eventdriven process chain(EPC) diagrams as seen in figure 45. The EPC diagram is the core diagram in which many of the relations are defined. The business rules is at Cim level a part of the EPC and ontology. To precisely describe a business rule and relating it to a decision, ARIS is using a familiar spreadsheet-like interface and connecting the business rule to the process in the EPC-diagram. To model the organization and its units, positions, roles and resources, ARIS provides an organizational chart diagram, which objects are available in the EPC diagram. They could be connected to the process with various types of relations. All the models are in the repository, supporting the design of a service architecture. If you try to make a diagram or an object with a name that is already existing, ARIS notifies the designer and warns him that the concept already are present in the repository. This is ensuring semantic consistency in the domain vocabulary and supporting reuse of concepts, processes, services and diagrams. The Service Browser in ARIS SOA Architect makes it easy to identify services for automating business activities. Because services are linked to business activities in the ARIS process model, executable BPEL processes can be generated automatically. After the BPEL processes have been exported, they can be executed on platforms, such as, IBM WebSphere, Oracle SOA Suite, BEA WebLogic, and SAP XI. The tool support that ARIS provides are some of the better seen in the market. 2 points.

32 REQ9: Holistic view

The holistic approach that exists in ARIS is a result of an integrated tool architecture, where models representing different aspects of the architecture are together contributing to a holistic view of the enterprise architecture. The tool also is integrating the process or service lifecycle phases like analysis, design, implementation, monitoring and test. There is also tool support down to various execution platforms by transforming the models to BPEL. One thing that is not existing in ARIS platform is an automated process from the business motivation model to top level services or processes. 2 points.

33 ARIS summary table

This table sumarizes the points given for ARIS.

Requirement	Key elements	Value
Service identification techniques	Bottom up	2
	Process	2
	Capability	2
	Collaboration	0
	Use case	0
	Business Service	0
	Architectural aspect	0
Service attributes	VAC, EPC and BPMN	2
Notation and language	Integrated models	1
Information model	ERM and UML-class	2
Service description	Access diagram, EPC,BPMN,VAC	2
Transformations	java scripts, EPC to BPEL	0
Process	Phases, disiplines, roles	2
Tool support	ARIS platform	0
Holistic view	Integrated Aspects and layers	1

Table 2: ARIS summary table

Part VI

The analysis of OASIS SOA works

In this part the OASIS SOA works is analyzed against the proposed requirements from part 13. The OASIS offers a set of elements that is quite comprehensive. First there is a reference model which defines the ontology for SOA. The reference architecture evolves the reference model into a broad framework for SOA. The OASIS blueprints methodology try to describe the steps, knowledge, disiplines and roles for identifying the core business services, participants and motives. It is reffering to the reference model, but is not all consistency in terminology. For instance why is it using the term actor(RUP) when the reference model talks about stakeholders and participants, or the use of term collaboration instead of joint action which is defined in the reference architecture. When it comes to tool support the methodology does not focus on or provide such facility's. The blueprints is all together vendor independent and neutral. It does mostly exclude descriptions and models for describing a PSM level artifacts, however it does include some models containing message patterns, policy and contract mechanisms and mechanisms for attaining visibility among others.

34 REQ1: Service identification techniques

The goals aspect is explored in the Needs and capability model. The motivation for participants interaction is the satisfaction of needs. From a consumers view, interacting with a service it is all about achieving business objectives. Further on the business objective are often related to the participants role in the social structure. The need and capability model shows the strong relation between needs and having them satisfied by some service provider with capabilities that match the needs. When talking about need and capability in this context, need is a concrete requirement a service participant is seeking to fulfill through a real world effect. Capability is a resource that is able to achieve a realworld effect on the behalf of the service consumer. Both need and the effect of capability are expressed in terms of a state. A service with a capability has a realworld effect that changes a state. The change in the same state also represent the fulfillment of some need a service consumer may have. So when modelling service architectures the service hierarchy may be derived out of the service provider's and consumer's needs and capabilitys. The needs must therefore be a measurable requirement that a service participant is seeking to satisfy. The way this SOA reference architecture talks about needs and capabilities are intuitive and pin pointing the essence of services. The OASIS methodology also has the notion of categorizing services. The core services, or level 0 services are identified first, and could be considered as a area of it's own rights so that a replacement would have minimal impact on other services. Then the support and shared services are identified. The level 0 services are the core services, the services that lies within the main value adding areas of the enterprise. The level 1 . . . n services are classified as virtual, support and shared services. The virtual services are used when one or several internal services are composed to deliver a view to an external actor or participant. There is some mixed signals here. The services that are exposed to external participants, especially the services that communicates with customers are obviously core to the enterprise. This is a point where the enterprise is exchanging values and has directly effect on the bottom line. Even thus the externally exposed services are just a facade and do not incorporate any business logic one may argue that they should be developed at the same time as the core services. They could maybe be designed even before the core services to address business motivation requirements. There is also the support services and shared services. Support services are often low granularity services and supports other services. They are often consumed by business services and span over multiple domains. Shared services are services that are common between multiple business areas. This could be various data services, for instance that a data service could provide other services with participant information. Business services are the services core to the business but not necessary level 0 services. They are likely to be consumers of support services. This classification is not very clear. How the services classifications are defined is not obvious. For instance there is no

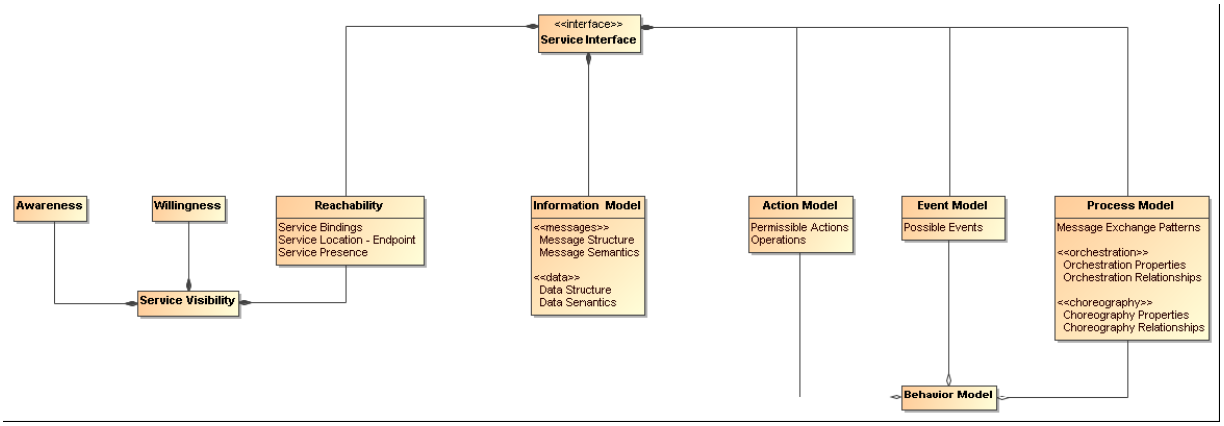


Figure 24: Service Interface model

definition of what a business service really is. Capability driven 2 points. Architectural aspect driven 1 points.

35 REQ2: Core service attributes

The reference architecture provides several models to cover this area at the CIM level. The concepts modelled are describing these and their relations to participants, events, capability's and realworld effects. For the business services the Stakeholder and participant model provides the necessary concepts for describing these. For more detailed tasks the Acting in a social context model is available. There are five different sub models of the Acting in a social context. These are: the Actions, realworld effects and events, Social actions, Interaction as joint action, Semantics and communication model and last the Transaction and exchange model.
PIM level models: The models in realizing SOA view fits with the PIM level abstraction level well, but one may argue that some of these models are also overlapping within the CIM level view. The behavior model is aggregated of the event, process and actions models of the service interface model, see figure 24.

Some of these models may be overlapping the service description requirement. This fits the requirements and even more. 2 points.

36 REQ3: Notation and language

Notation is not a issue that is in the focus of the reference architecture, but when taking the language in consideration there is a wide variety of models to explore in the reference architecture. The models span over the CIM and PIM where the PIM level diagrams and concepts are mostly refinements of the CIM level. There are also some vague descriptions of a profile in the blueprint methodology for representing the big picture. This includes a service hierarchy, it's major related associations, interactions and participants. Many of the models interleave the aspects defined for the research requirements. The concepts are well adapted to the world of business and as the technical world. However even if the notation has been mentioned and UML profiles have been proposed but not formally described. This has obviously not been an area of focus. This will give the OASIS papers a 1 pointer at this requirement.

37 REQ4: Information model

For the information model at CIM layer the reference architecture provides a resource model see figure 64 that connects resources to stakeholder and defines ownership of resources. Together with concepts defined in domain investigations like the Event or stage 0 deliverables from the OASIS methodology, it will contribute to the representation of the ontology. Also contributing to the ontology is other concepts discussed and defined in the process of documenting the business.
PIM level models: The models in realizing SOA view fits with the PIM abstraction level well. At the Ontology and information model aspect is modelled by a part of the service description model in service interface element. See figure 67 and 68. This is a partly a refinement of the resource model as the general description class is a subclass of the resource concept. The service description class is again a subclass of the general description class. The ontology that is represented at CIM layer is defining how to talk about the resources and other modelled concepts. This is good support for information modelling. 2 points.

38 REQ5: Service specification

As mentioned in requirement 2 above some of the concepts and constructs that may belong here is present there. But the acting in a social context models are having concepts for modelling a behavior model with actions and events. In the more detailed Service description model there are the Service Interface, behavior and functionality models that seems to express all that we would expect of a service architecture. In the model showing relationship between, actions and service descriptions and the service interface model, most

of the required concepts are described. The concepts are preconditions, interfaces, message, behavior and contract. The one I could not find where the post conditions but they could maybe be expressed as events 25. The concept of endpoint is analogous to a WSDL2.0 interface operation.

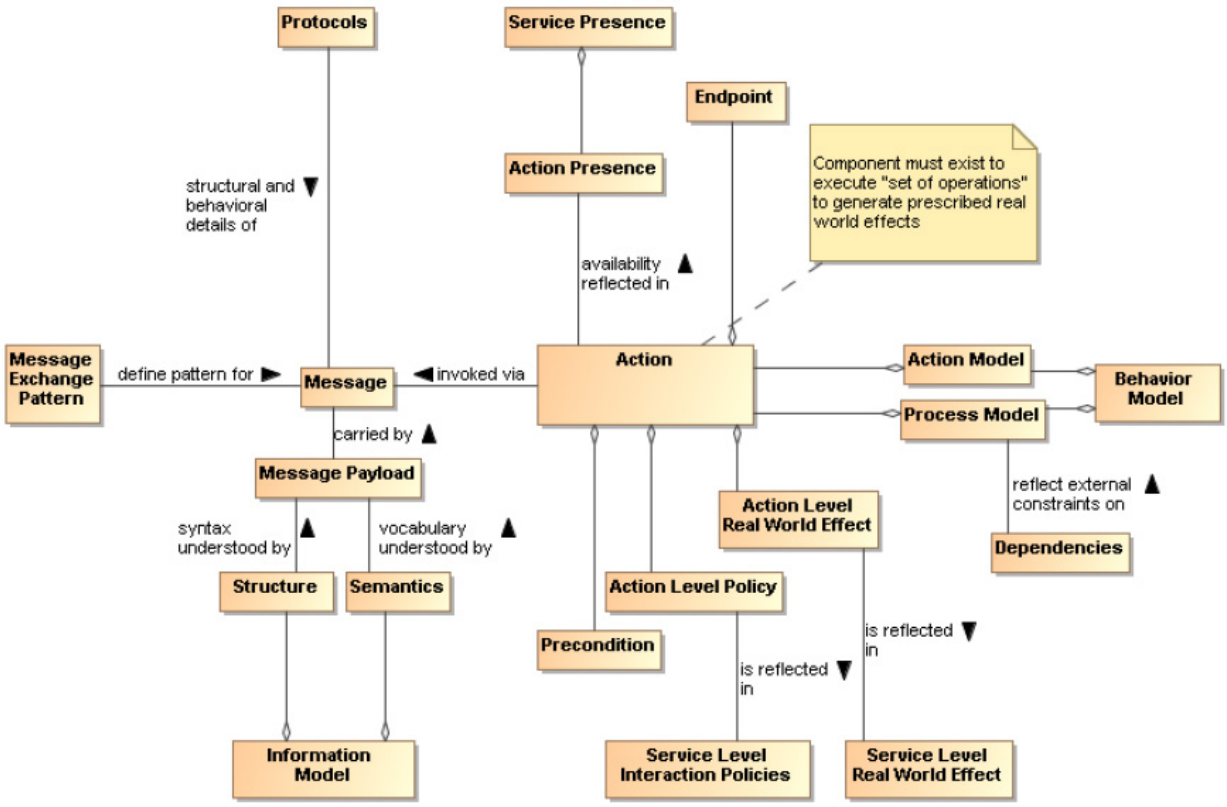


Figure 25: Relations between actions and service description

This is a rich language to express services and their environment and gives the methodology 2 points. Whether the OASIS reference model is used only as a model at CIM level or a full model of both CIM and PIM, it has the ability contain the information needed for describing the service oriented architecture.

39 REQ6: CIM2PIM transformation

The transformation is not an issue in this context because this reference architecture does not provide two distinct metamodels to transform. In fact

neither the reference architecture nor the methodology does talk about transformations. 0 points.

40 REQ7: Process support

The reference architecture has a methodology that is mentioned in state of the art part. The methodology is only a draft and may not be a part of the official OASIS reference documents but it is assumed that it is in this document. The process described in the draft is a "cookbook" style prescription for finding which services to design. It has a 3 phase lifecycle. First the stage 0, then the event and at last evaluation and deciding next steps. It has descriptions for how to play the different roles in the process and who from the organization that should fill the roles. There are also descriptions of what should be produced during the activities. This contribution to the methodology is very down to earth and is very specific about how to proceed. Even thus the methodology is not completely conformed with the reference model and reference architecture this is not a bad process description and I will give it 2 points.

41 REQ8: Tool support

The tool support is not an issue for the OASIS blueprints. This because it is that I know of, non existing. 0 points

42 REQ9: Holistic view

The OASIS documents that is taken under consideration in this paper are embracing a holistic view on the service architecture and illuminate the aspects required, and are seeing the architecture both in the context and the contexts needs. There is a big hole where tool support should have been, and the OASIS methodology is a kind of unfinished or uncomplete.

43 OASIS summary table

This table sumarizes the points for OASIS works.

Requirement	Key elements	Value
Service identification techniques	Bottom up Process Capability Collaboration Use case Business Service Architectural aspect	0 0 2 0 0 0 1
Service attributes	Stakeholder and participant model, Acting in a social context	2
Notation and language	Rich language and notation	1
Information model	Resource model,	2
Service description	Acting in a social context, Service description model	2
Transformations	non existing scripts	0
Process	OASIS blueprints	2
Tool support	not known	0
Holistic view	Aspects, layers, tools, process	1

Table 3: OASIS summary table

Part VII
The analysis of ArchiMate

In this part Archimate is analyzed against the proposed requirements from part 13.

44 REQ1: Service identification techniques

As we see in the figure 60 the language allow us to describe the customer and how he has a need or wants something that gives him a value. In this case he wants security in form of an insurance. So he wants to buy a product. The Business services and application services are grouped together to form a product which correspond with the value requested by the customer. The product can as already mentioned, contain both Business and Application services. The purchase of a product gives the right to use the services. Together with the product is a contract that describes the terms for using the product and the services. This includes characteristics, rights and requirements associated to the product. It is a Product-value driven service identification, 2 points. Archimate has the business service concept which is modelled as first level services supporting the underlying business processes,

but this is not described in documentation, 1 points. Archimate has also the notion of collaboration but this is not described in ArchiMate as a source for identifying services but the possibility is there, 1 points. There is in ArchiMate some aspect classifications and they follow archimates layer architecture. This gives business services, application services and infrastructure services. 1 point because this is not described formally and explicit in the documentation.

45 Req2: Core service attributes

The ArchiMate language is not a very rich language. For modelling the service architecture at top level, the ArchiMate is right on. It has a clear distinction between business services an business processes. The Business services that are compiled to a product, are realized by business processes. The Application services are supported by Application functions or interactions. The roles played by participants are expressed by having Business roles played by an actor, and roles can interact through a business collaboration. See figure 61. 2 Points.

46 Req3: Notation and language

The notation an language are really simple and are well adapted to their purpose. Both business people and technologists can relate to this notation and language. The product and service diagram in figure 60 is really intuitive and does not require any indepth knowledge about business vocabulary or technical economics. The fact that both the business layer model 61 and the product model 60 are kept simple makes it a great tool for bridging the gap between the business and technology domain. 2 points.

47 Req4: Information model

To represent the information model ArchiMate has the notion of information. The information aspect has at top level a meaning. At business level the information is represented by Business objects, data objects at application level and finally artifacts at technology and infrastructure level. See matrix in figure 58, the information row. The information aspect is represented at every level from the semantic meaning down to the physical representation. 2 points.

48 Req5: Service specification

For a deeper specification of services the language and notation are a bit to simple. It is described no way of modelling the messages and more complexed

interactions. Neither is it possible to describe pre and post conditions nor the specification of interfaces. 0 points.

49 Req6: CIM2PIM transformation

Since the ArchiMate is a language and it is up to the vendors how they wish to implement the language. However the ArchiMate has been fully integrated into the ARIS platform. There is not much information about transformations and ArchiMate available from ARIS. But since it is fully integrated there is possible to write the transformations your self in the ARIS transformation script facilities. See 29. 1 points.

50 Req7: Process support

The processs support is not an issue because it is not specified any specific, but there are some references to TOGAF[29] among others. UML can be abstracted using ArchiMate or in other words UML could be the notation of Archimate. TOGAF can be typified with it. There is a high coherence between TOGAF views and ArchiMate domain viewpoints in certain areas. TOGAFs Business Architecture component can be visualized with ArchiMates business viewpoint concepts, Information System Architecture with application concepts and Technology architecture with the concepts of infrastructure concepts in ArchiMate. The process support is not predefined but the aspects of language is coherent with other existing process frameworks. 1 point.

51 Req8: Tool support

Like mentioned in the REQ6 ArchiMate has newly been integrated into the ARIS platform, and has thereby tool support. There are also other tool vendors supporting the ArchiMate language. One is the Bizzdesign Architect. This is the tool that was used to draw the diagrams used to illustrate ArchiMate in this paper and also the book [21]. Another is the Avolution Acabus. The tool support do not get much better than this. 2 points.

52 Req9: Holistic view

The ArchiMate does promote a simple but holistic view of Service Architectures. It covers all the layers from CIM to PSM and even the physical infrastructure. Archimate has three aspects cross cutting the layers. These are the aspect of information, behavior and structure. 2 points.

53 ArchiMate Summary table

This table sumarizes the points for Archimate.

Requirement	Key elements	Value
Service identification techniques	Bottom up Process Capability Collaboration Use case Business Service Architectural aspect	0 0 2 1 0 1 1
Service attributes	Behavioral business and application concepts	2
Notation and language	Simple language and notation	2
Information model	Meaning, Business and data objects	2
Service description	Behavioral aspect and contract	0
Transformations	Decided by the tool support	1
Process	Flexible but TOGAF is referenced	1
Tool support	ARIS, Bizzdesign Arhitect, Avolution Acabus	2
Holistic view	Aspects, layers, tools, process	2

Table 4: ArchiMate summary table

Part VIII
POSI, a Unified approach

It is at this stage that all knowledge found in the state of the art technologies part above, are to be accumulated into a guide through the various methodologies and techniques for identifying services in a service oriented architecture. So how approach this challenge? There are two main strategies for achieving this. Either by describing a rigid process with a fixed chain of events. Alternatively by offering a tool box of techniques where languages and processes that can be composed according to the context of the product to be developed. The first strategy is to firmly lead the development through a predefined end to end process with an unambiguisly language and notation. This gives the stakeholders a predictable environment for the development of

the product. This could be an advantage with respect to forseeing risks and gaining trust among the stakeholders that are participating in the development process. People tend to feel more comfortable when the number of unknown factors are keept to a minimum . It is important that the stakeholders that are sharing information, are able to freely express their knowledge and communicate within the group. However the proposed approach may not fit the current challenge or context. The business might for instance already have a lot of money invested in tools and resources for documenting and modelling their services or end to end processes. Then deciding for a methodology that do not integrate the information already acquired, might not be a good idea. The strategy based on a methodology including a wide range of interleaving techniques, phases, tools, languages and notation, is offering an agile way of thinking service identification. This also corresponds well with the agile philosophy of SOA. This means that the business can tailor their development process and align it to their business strategy. Then the models can be exported to existing execution platforms as BPEL. Then these resources should of course be taken in to consideration. The methodology used in this case should take advantage of this fact and be fitted into this context.

54 The POSI philosophy

The POSI method is a unification of the state of the art technologies and is trying to combine and reuse the good elements found. The process has elements from both OASIS and COMET-S. The need capability model is taken from the OASIS reference architecture. From COMET the notion of Collaboration and use case description. The bottom up technique is taken from ARIS. From the ISO 19119 [30] that is also used to some extend in COMET-S, the perspective oriented approach. The service perspectives are used to categorize the services and to outline the architecture. This perspective oriented approach, is using a derivation of the six perspectives of seeing services from ISO 19119. It is added one additional perspective that in addition to the other six perspectives should be present, namely the rule services. This perspective could be a part of the management services but are isolated due to it's significance. This gives 7 perspectives in all.

1. Interaction services are boundary services that focus on presentation of media documents, information, management of user interfaces and giving users access to workflow services. The interaction service can enable the degree of user control with respect to controlling the chain of events. The interaction services does not differ between human or system interaction. They represent the contact points between the service architecture and outside participants. These points of contact can be defined and identified by one of the service identification techniques and is representing the areas where the business is exchanging values and

information. May for instance be seen as a product in the Archimate language.

2. Process services are services for support of specific tasks or work-related activities conducted by humans. These services support use of resources and development of products involving a sequence of activities or steps that may be conducted by different persons. These services are categorized in business and application processes.

3. Model/Information management services are services for management of the development, manipulation, and storage of metadata, conceptual schemas, and datasets.

4. Rule services. Services for automating decisions, especially supporting workflow decisions, configuration and management.

5. Processing or function services are services that perform large-scale computations involving substantial amounts of data. Examples include services for providing the time of day, spelling checkers, and services that perform coordinate transformations (e.g., that accept a set of coordinates expressed using one reference system and converting them to a set of coordinates in a different reference system). A processing service does not include capabilities for providing persistent storage of data or transfer of data over networks. These services are not in the scope of this thesis.

6. Communication services are services for encoding and transfer of data across communications networks. These services are considered not in the scope of this analysis.

7. System management services are services for the management of system components, applications, and networks. These services also include management of user accounts and user access privileges. These services are not considered within the scope of this analysis.

These perspectives are as mentioned the used to categorize the existing services, it is also used to identify the different services. See the POSI for SOA identification process in figure 26 that outlines the development process. However the steps from processing service to management service identification is not covered in this analysis and appear in grey colored process boxes. The two long boxes are processes that are executed in parallel with the others. There are an exchange of information between the processes that runs simultaneously. Also in the lifecycle diagram is the different service identification techniques described and linked to the process they are applied. As seen in the SOA identification process figure, some of the processes is combining several of the service identification techniques below.

- Process

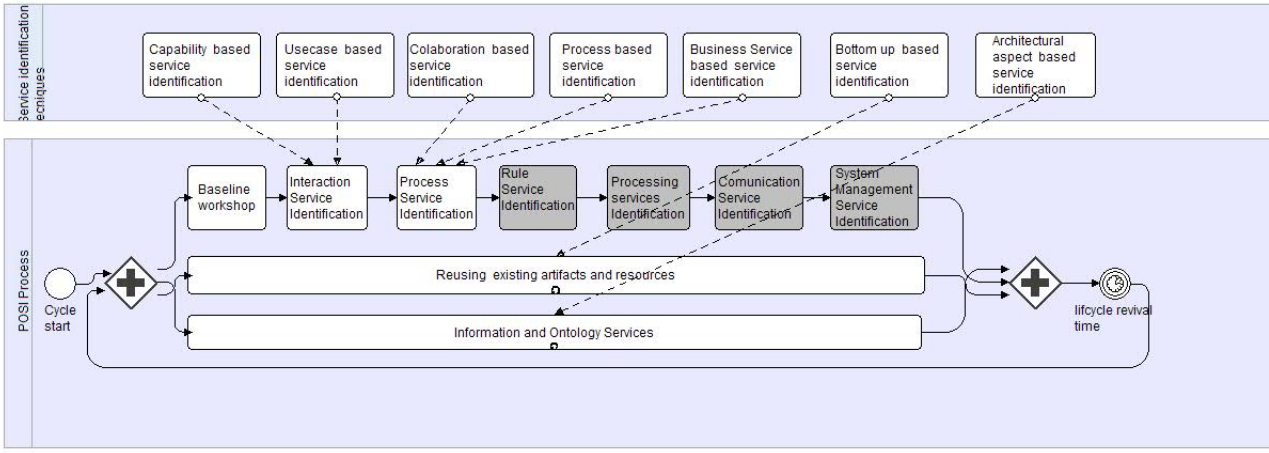


Figure 26: The POSI process

- Capability
- Collaboration
- Use case
- Business Service
- Architectural perspective
- Bottom up

55 POSI methodology

The starting point is what kinds of internal or external resources are already present within the organization and it's context. When taking the development project in consideration, before the service identification process can start there has to establish a base line for the development of the architecture. The baseline could be a collection of documents describing the existing resources, knowledge and other elements that could be of interest understanding the context and core mechanisms of the business. Things of special interest are documents describing the organizational structure with roles , business processes, business rules, existing architectures, tools, strategic market and economic analysis. Also interesting is of course all the documents concerning the development project. A problem might be to involve external stakeholders in the process and be able to handle confidential information with

respect at the same time. POSI is executed by a series of workshops each taking a different perspective. For the results to be good it is important that the workshops becomes collaborative activity [31] and that the stakeholders contribute with their knowledge in the discussions to make a common understanding of context and problem.

Roles: The size of the event team is dependent on the size of the project. But a team's roles would typically include a:

- Project manager for delivery, facilitator
- Lead business analyst
- Lead architect
- Administrative assistance
- Technical support
- workshop co-ordinator

The event is lead by the Project manager or also called the facilitator. The facilitator should have these capabilitys.

- Strong communication skills
- Strong listening skills
- Broad understanding of Business domain
- Ability to coordinate teams
- Ability to lead a discussion that delivers
- Must be experienced

55.1 The baseline workshop

The baseline workshop is about mapping out the existing picture of the business and it's context. The as-is picture. Before any activities begin the initiate taker or project owner must gather all accessible documents and information related to the project. Organize those who are relevant according to the perspectives described in section 54 for use in the process. When all documents necessary are collected, the project owner or initiate taker along with the business architect, application architect, the responsible for technical resources and human resources will have an initial workshop. If there is allready an existing service architecture the services must be documented and categorized according to the service perspectives. This is because it makes it possible to reuse the existing services when they are classified and described.

Another important baseline activity is mapping out the organizations resources, stakeholders and their knowledge. It is crucial that the stakeholders and their knowledge is considered according to their role in the business and the possible role they will play in the process. It is also central to the fact that the knowledge the stakeholders inhibit will form the development process itself. If stakeholders that are central in the development process are experts in SCORE analysis and BMM modelling, you would like them to conduct the goal modelling together with the architects. This survey will answer the question -which stakeholders will attend to the Service identification event? Who has the right authority, knowledge for filling the roles and the time to take part in the process? As the project starts to evolve, it is important that vocabulary used to talk about the domain is documented. Alternatively finding a predefined ontology for their area of business. In addition a collaborative work supportsystem should be set up to share informations and documents. Another issue for this event that needs to be settled, is what tools and languages to use in the SOA development process and management.

Work products of the workshop: Organization model. Consists of a diagram including the roles, resources and positions in the organization. In addition a matrix showing the resources special knowledge areas, authority and properties related to roles or resources needed in the development process. Project plan showing the sequence of activities, people that are responsible or participating and resources needed to achieve project milestones and objectives. A categorization of all project documents, sorted and arranged by the perspectives used to identify services and existing services if any.

55.2 Main service domains as Interaction services

The interaction services are boundary services outlining the service architecture and the communication with external participants. The baseline workshop provides the development starting point for the interaction service workshop.

55.2.1 Process

The main objective for the first workshop is to create the big picture outlining the service oriented architecture and the service taxonomy. A big picture will provide an overall guide to the enterprise, or a project, and will give foundation for organizing the capabilities of the organization into services. Finding the core services can be about keeping the view of the business as a whole and analysing the communication to external participants. These are the points of contact where values are exchanged. These are the interaction services exposed to the participants. Depending on the kind of product market analysis done, if any. The analysis should indicate what to offer to whom, and aiding choosing partners like financial institutions, shipping partners or

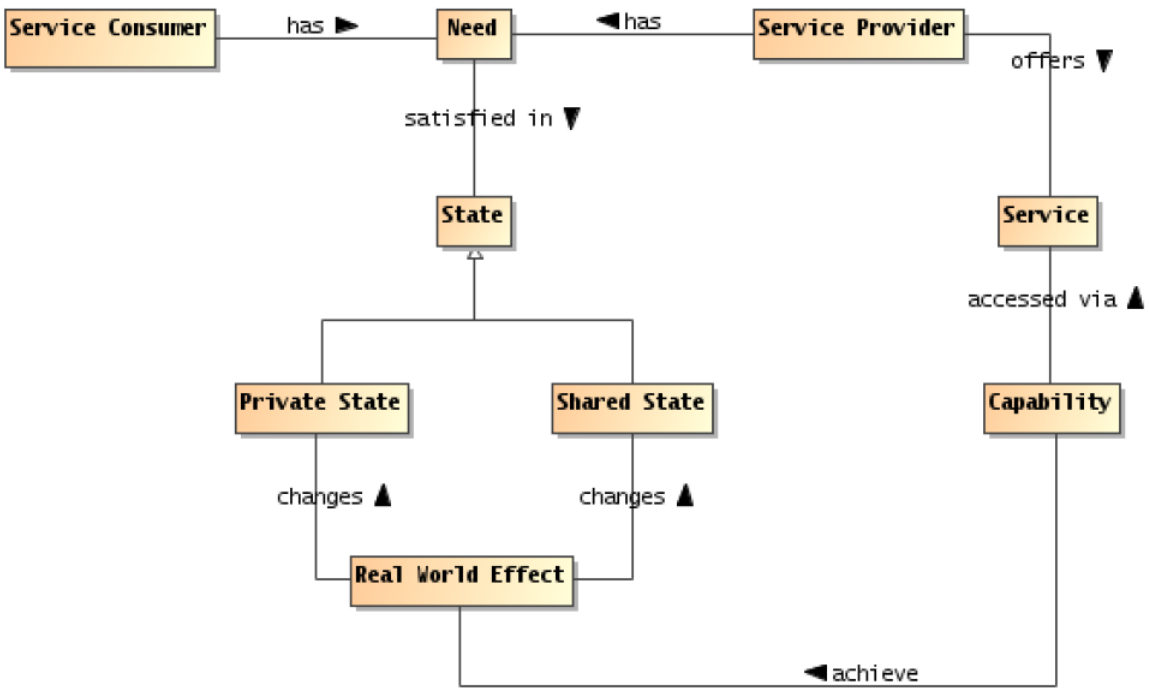


Figure 27: The Need and Capability metamodel

suppliers . These services can be identified by any of the described techniques in REQ1, section 13.1. To identify the main service domains a need capability analysis is applied on the external stakeholders or participant.

Workpackages from the the interaction service identification: A class diagram to express a hierarchy of needs, capabilities, participants and services. A description of the services and related concepts in the need capability model.

55.2.2 Language and notation:

The Need capability language used is defined by the OASIS Need capability model seen in figure 27.

The need and capability model is used with a simple class diagram using stereotypes to represent the concepts. An example of such is seen in figure 30.

55.3 Process services identification

This is the third workshop and is focusing on the services that expands the interactions services into the process services. The process services are of two different sub types, business services and application services. The notion of business services in Archimate is a good way of differencing services with a human interface to services with system interfaces. The service is in this context defined as an unit of functionality that some entity makes available to its environment, and it provides some value for its service users.

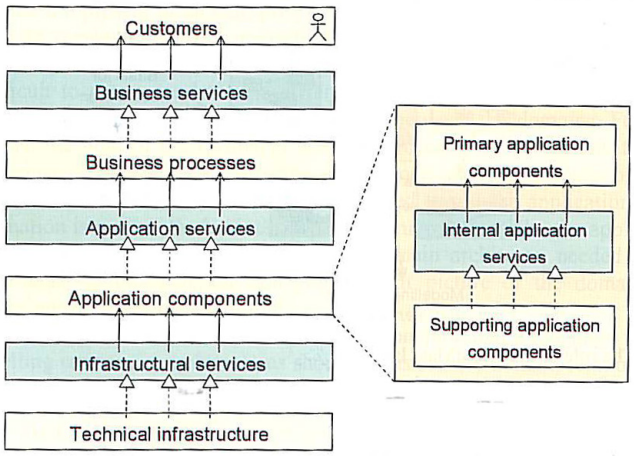


Figure 28: Service layer stack

Service orientation typically leads to a layered view of the enterprise architecture, where the concept of service is one of the main linking pins between the different layers. This will lead to a stack of service layers and implementation layers in Figure 28.

55.3.1 Process

The interaction services found, are detailed with use case descriptions denoting the underlying process services and are documented using the use case template in table 5. This particular template is taken from the COMET-S methodology[4]. As seen in the use case template there is two diagrams needed, a use case diagram and a behavior diagram. This process is parted in two steps or iterations. The first iteration is identifying the services as new use case realizing the interaction services. It is done by making a use case description of the interaction services using the use case template. This step is combining the notion of a collaboration playing out the participants and

Table 5: Use case template

Part	Description
use case id	The identification of use case x for later reference
UML use case diagram	The use case diagram including extends and includes
Goals	A description of the goals or capability's of the use case
Actors,Roles	A textual description of roles in the use case and their resposibilitys
Textual description of scenarios	The use case scenarios expressed in some behavior diagram. A BPMN diagram in this case.
Pre and post conditions	A description of the use case pre and post conditions.
Non Functional Requirements	Description of non functional requirements using text

services roles, and using use case descriptions and scenario model to express the decompositions of the interaction services that arises. To show how they collaborate their behavior is modelled in BPMN. The process modelling of the collaboration will in addition to showing the services relations also uncover the need for additional services. This means we are using a process based service identification technique. Now the level 1 services are identified. If the service granularity is not yet fine coarsed enough, another service discovery must be done for identifying the level 2...n services. To identify the next level services the same exercise is done again, only moving use case focus one service level down. If the granularity is suficcient fine coarsed the service operations must be identified. This second part is expanding the newly found process services and another use case analysis is conducte. The behavior model that is describing how the services behave and interact are now used to identify the service operations.

Process service Workpackages: Use case descriptions of the interaction services and process services including a use case diagram showing the interaction services, process services and the internal and external participants. Also included is BPMN diagram's showing the services behavior, how they relate to each other and the messages sendt between them. These use cases may represent several layers all depending on the complexity of the service architecture.

55.3.2 Language and notation

The decision to use a use case diagram for further identifying the services, was made because it is an intuitive and highly comprehensive way of breaking down the functionality of the services. In addition the participants and services can be seen as roles in a collaboration and services occur as the roles are played out to meet the needs with the service capabilities. The diagram is also easy understood by business people. The use case diagram is also used with stereotypes from the OASIS need and capability model from figure 27. The notion of business services and application services is not explicitly expressed in the examples with any business service stereotype or with a dedicated symbol, but may be identified by it's internal participant. If the internal participant is a human participant the services is a business service compared to application services that are system functionalities that can stand alone and communicate directly to external participants or support business services. For this particular use case, the diagrams are focused on the part of the architecture that is to be automated by a system. So most of the services identified in example are application services except the "ship order" service that has an internal participant. See a part of the evolving SOA in figure 29. A bigger diagram may be found in the appendix figure 78.

It is now easy to evolve the service taxonomy and expand the process services that are realizing the interactions. If internal human resources are involved in the service behavior, they will be modelled as any other participant. They will stand lane by lane in the BPMN diagram, but the business service is as a service with a human Participant. This means that an individual service can not be both human and a system. So for generating for instance an interface specification there has to be a way of differing human from system services so that the communication between them are added correctly to the interaction services. The use case diagram evolving services in figure 29 shows the breakdown of the service domains. The diagram should also be used to connect the internal human participants to their related services. In this way if the current service requires the attention of internal human resources, they can be added to the behavior diagram. The snapshot of the diagram shows the architecture under development. Not all the concepts from the need capability model are included due to readability, and not all internal participants are yet included. Also the messages to and from services, and the service processes including the data objects related are added in the BPMN diagrams. The behavior is expressed through the BPMN language which has four distinct categories of concept elements.

- Flow Objects: Events, Activities, Gateways
- Connecting Objects: Sequence Flow, Message Flow, Association

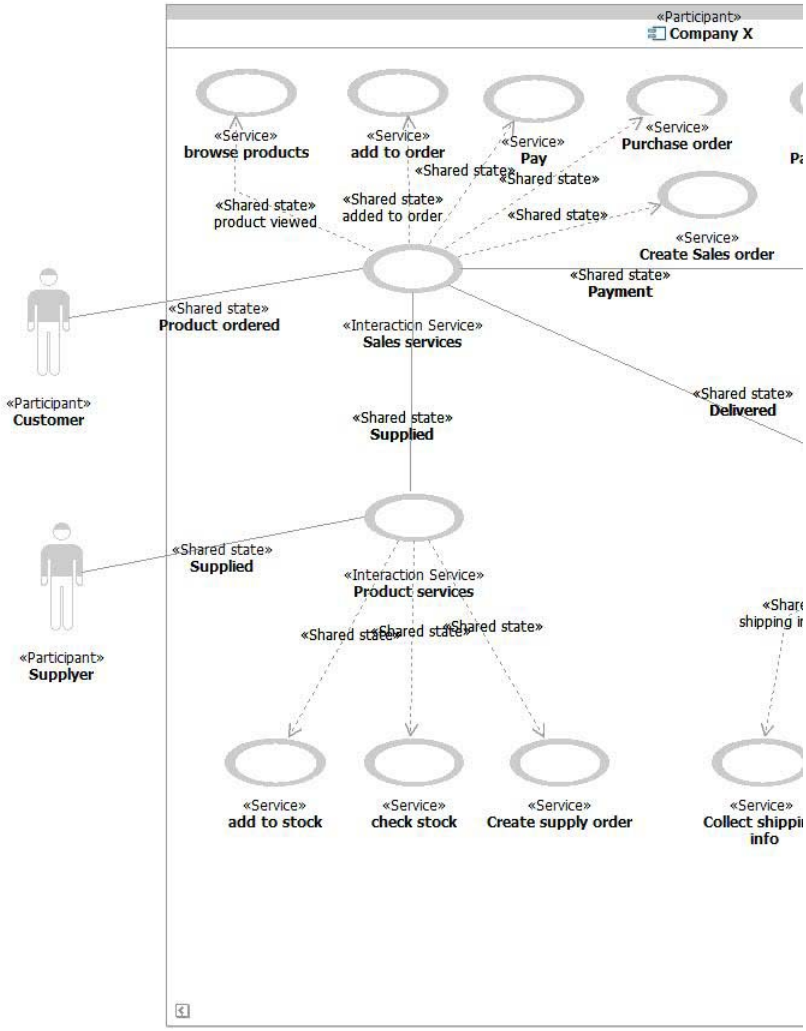


Figure 29: The evolving SOA

- Swim lanes: Pool, Lane
- Artifacts: Data Objects, Group, Annotations

55.4 Information service identification

There is two parts of the development process that go on parallel with the other service identification activities. This is the modelling of the information model and identification of information services, and reusing existing resources and artifacts. These activity's are a part of all the service identification workshops. The information model is an ontology including semantic models that represent the meaning of the concepts and a class diagrams representing structure and inherent properties. The ideas for the identification of information services and information model is taken from the COMET-S methodology[4].

55.4.1 Process

A use case model is a good starting point for identifying types and structures in the information model. The actors/participants and objects from the behavior diagram are excellent candidates for being information types. All pass and return arguments between participants and/or services are also type candidates. Excluded are native types like integer, double and boolean. Included are the description of the service interfaces, both internal ones and those presented in the interaction services. The elements in service information model representing objects and behavior is merely conceptual. The questions to ask are what objects and values are communicated, who are communicating, and how they are communicating through operations.

The base for concepts in the information model is the domain of the business and it's context. Therefore domain expert knowledge is extremely important when defining the information model. To identify the types it is common to pick out the nouns and subjects within the area of concern. Example of information type candidates are:

- Participants
- Person
- Organization
- Group
- Role
- Services
- Systems

- Documents
- Resources inherit and used by the architecture
- Information (e.g. a sales order)
- System products

Also constraints should be investigated and formalized through a textual description and a formal constraint language. Workpackages: UML-class diagram with OCL, including textual descriptions for each element.

55.4.2 Language and notation

These complex types (entity-types/data-types/features) are modelled using UML class diagrams with relevant operations, attributes and associations. Constraints related to data objects are expressed using UML, more specific OCL(Object constraining language).

55.5 Reusing existing resources and artifacts

When reusing artifacts and resources it is crucial that the categorizing of existing artifacts and resources done in the baseline workshop is completed. This part of the process is using a bottom up approach to identify matches between already existing services and new service requirements. There may also be information in the categories to cast light over issues in identifying business rules and, pre and post conditions or any other design issue during the SOA development.

55.6 Transformations to PIM

The structure conforms well with the SoaML service architecture model described in Appendix A figure 35 used in the COMET-S methodology. Roughly the transformations would play out like this: All services are mapped to services and or required services, participants to participants, messages and data objects maps out to the description of interfaces. Data objects and service attributes to information model.

55.7 Other service aspects

These other aspects, namely the Business rules services,processing services,Communication services and Management services are not considered as apart of the scope and might be a subject for further work.

Part IX
POSI applied on buyer seller example

In the following part POSI has been applied on the buyer seller example. The processes of identifying the interaction services, process services and information services are conducted and modelled. Some of the figures in this part are rotated to get the whole picture.

56 The case

As mentioned earlier in the thesis introduction, the case is a typically Buyer Seller scenario, a service oriented project with the object is to automate functions of the EA. The organization have decided to launch a webshop based daughter company that buy and sell both physical and digital products. The shop tries to do business in the business to customer segment, and is using web technology to achieve this. The scenario would play out like this. The customer will search through available products in the web shop. Then decide to add an sales order of available products. When the customer is finished he will confirm the order to the web shop. The order will then have status salesorder that is a unconfirmed order, and the web shop will instantiate the process for retrieving transport proposals and the prices for these. Alternative transport plans and prices are given to customer and he has to decide which plan fit his needs, if any. When a customer picks the plan, he submits his choice with payment information to the web shop. Web shop then process the payment information and if payment check is confirmed a shipping request is sent to transport company and the customer gets a confirmation that order is completed.

57 Interaction services identified

The need is satisfied through the real world effect sales as seen in figure 30. To achieve the realworld effect the company X has to have the capability to sell products. The name of the service might vary depending on the service identification technique used, but in this case it looks like the real world effect names seem to give reasonable names to the services.

In our buyer seller example we have the external participants customer, shipper, production and a financial institution like a bank. The customer

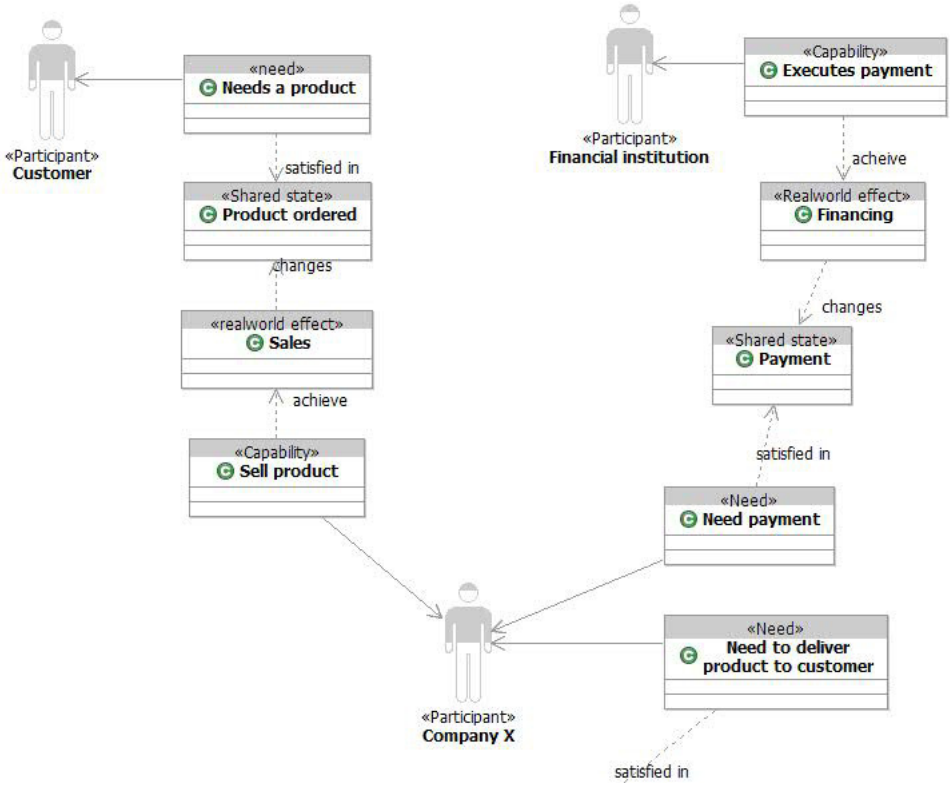


Figure 30: Need capability model

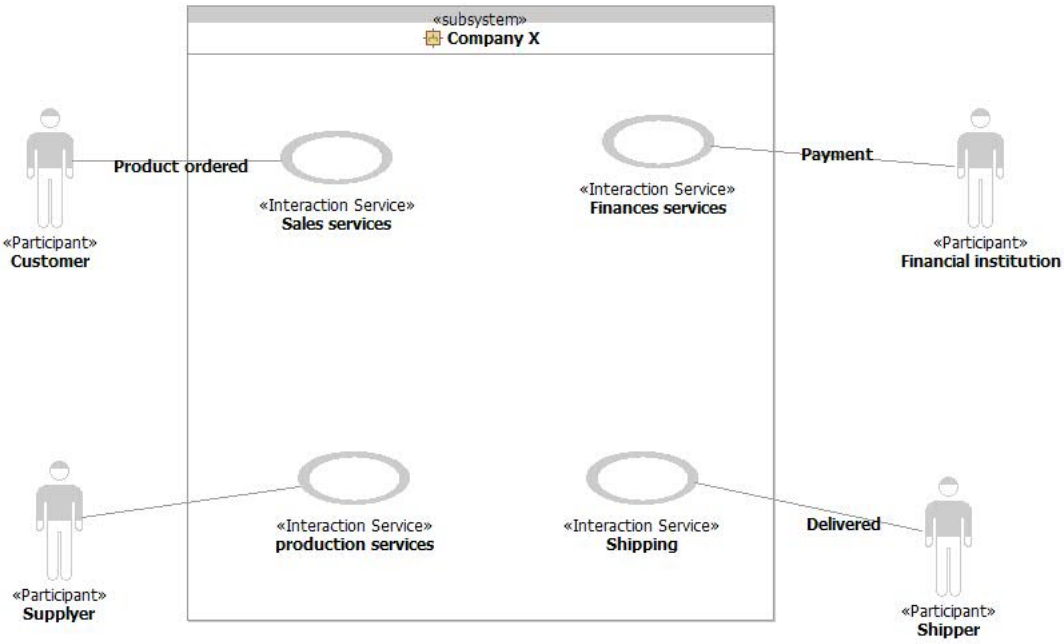


Figure 31: Use case top SOA view

which is an external participant is initiating the services by contacting the shop, sales, or customer service. He has the need to buy a product that requires an outlet, sales or customer service. To satisfy the needs, companyX has to have certain capability's. This gives companyX some needs that has to be met by appropriate capability's like the capability to offer financial, supply and delivery services. And then the top down core services, or services domains are identified. The top down services are the:

- Sales
- Finances
- Shipping
- Product

But to expand the services this kind of diagram becomes to crowded and the readability becomes poor. So the diagram is transformed into a use case diagram. The actors are representing participants, services represented by use case, the capability is represented by any connector. The concepts are denoted by stereotypes seen in figure 31.

58 Process services identified

The process services can be evolved from the Use case top SOA view by using the use case diagram as a collaboration diagram playing out the participants roles revealing the services. See the evolving SOA in figure 29 or in the appendix a bigger picture may be found in figure 78. The Behavior level 1 services showing the how collaboration between the interaction services plays out and identifying process services still not identified. The diagram is seen in figure 32.

And in the second iteration of process identification the level 2 service create sales order, is modelled in a BPMN diagram seen in figure 33.

59 Information services identified

To identify information types, the described data objects and services in the BPMN models is used as a source. The messages sendt are a product, a productlist, a selection and a salesorder. These concepts are represented in the information model together with services and participants information seen in figure 34. The model is reflecting only a small subset of the infomation concepts needed.

60 Other services

The other service perspectives are not within the scope of the thesis and will not be explored in this paper. These are the business rule, processing, infrastructure and communication and system management services.

Part X
POSI evaluated

61 REQ1: Service identification techniques

The service identification techniques used in POSI are the Bottom up 2 points, process based 2 points, Capability 2 points, collaboration 2 points and 2 points

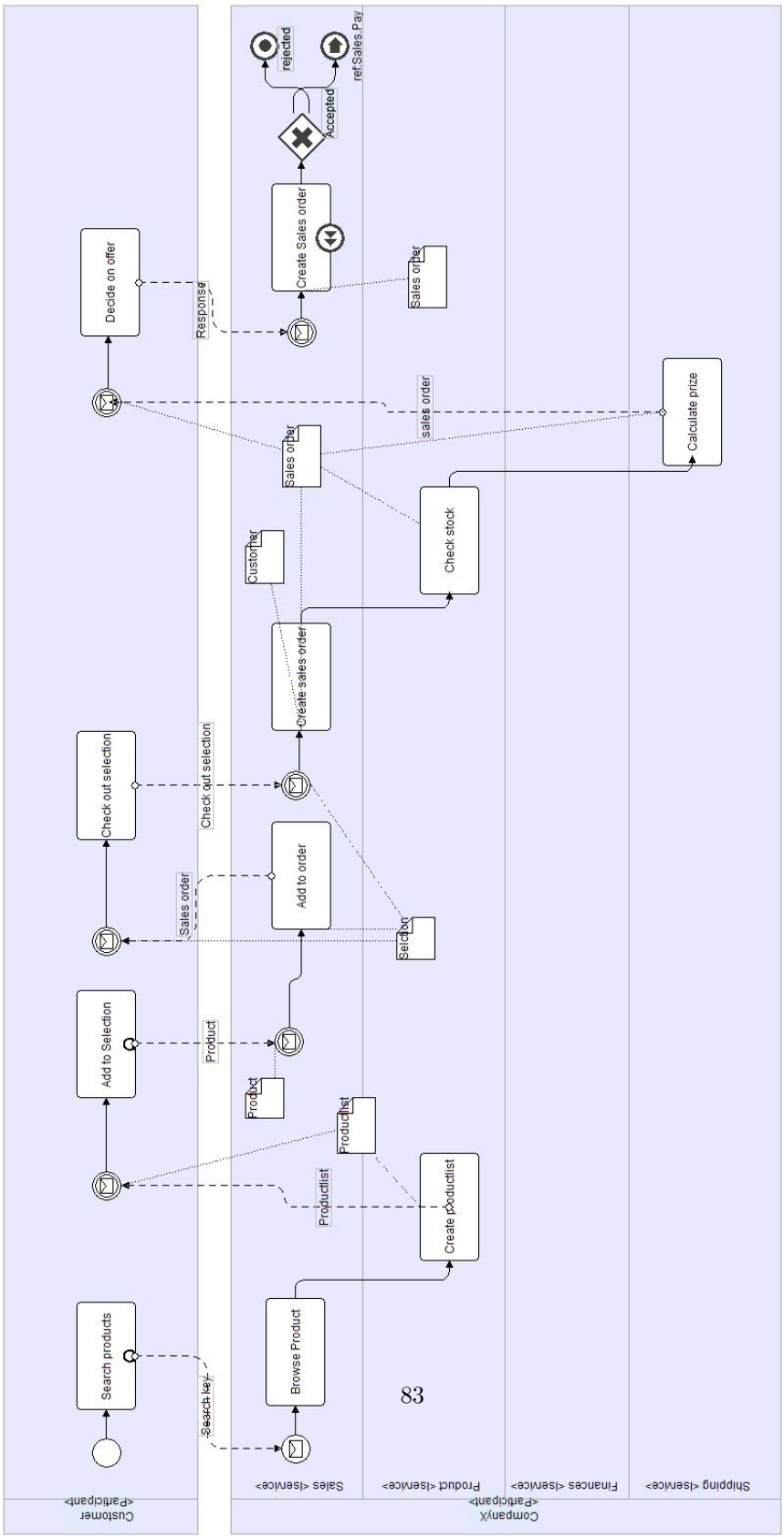


Figure 32: The BPMN diagram of the collaborating interaction services

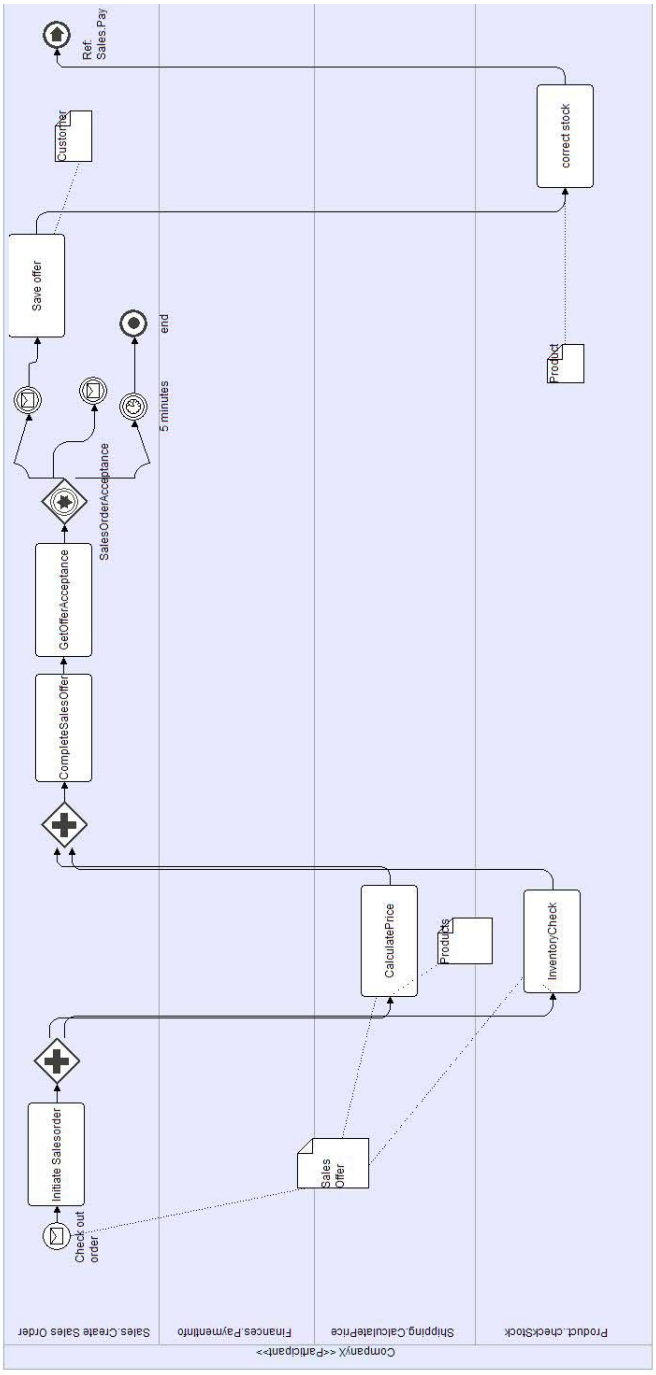


Figure 33: The BPMN diagram of the service: create salesorder

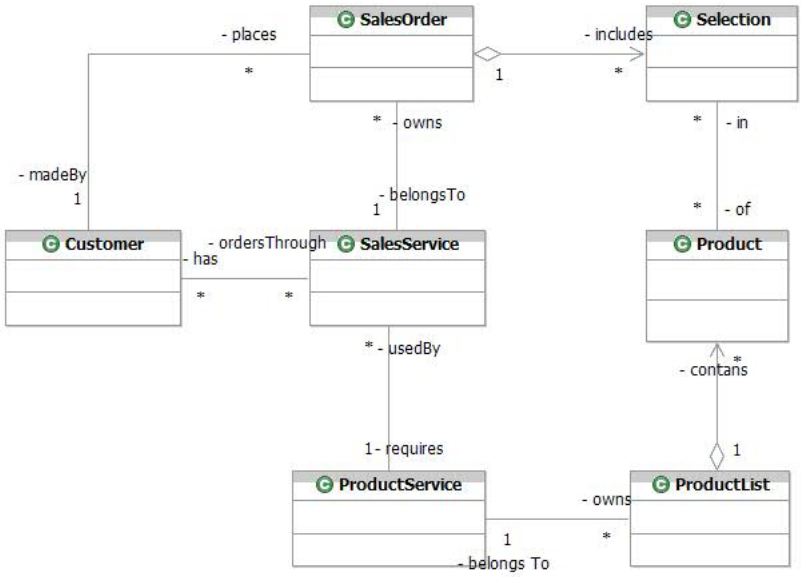


Figure 34: The Class information diagram for the service: create salesorder

for Use case driven approach. The different techniques that are used in POSI is already described in the requirement part 13 and will not be described further here.

62 REQ2: Core Service attrib

This requirement is about giving the service a reasonable and meaningfull humanly name and describing its associations and relations with other participants and services. In addition also the messages sent between them are denoted. POSI achieves a 2 in this requirement.

63 REQ3: Notation and language

The notation and language are a good match with the people modelling and the need for information describing the architecture. The use case diagram is a simple diagram that is not complexed and this gives little noise. The BPMN diagram is a very intuitive notation and language. It contains a clear and simple set of pools, process types, gateways, objects and events. 2 points.

64 REQ4: Information model

The POSI information model is a UML class diagram and can be linked to an ontology model expressed in OWL[28] or with an OWL UML profile. It could also be represented as a native OWL model. The information model and ontology are representing the vocabulary that is used to talk about the domain. Specially the participants, resources used, and messages sent including their arguments, should become entities in the information model. This results in 2 points.

65 REQ5: Service specification

This requirement includes a more indepth view of the service. The modelling language must be able to express possible many Service interfaces, service operations, the operations pre and post conditions, parameters and exceptions and a service contract including owned behavior. POSI has To generate the multiple interfaces with its service operations are done by looking at the messages going in and out of a service lane. Some of the pre and post conditions can be derived by the sequence of tasks or services or events. Services located before a certain service in the flow is a pre condition if it is not an optional operation or service. Parameters can be expressed as dataobjects attached to the message flow. Exceptions has its own symbol in BPMN and is included directly in the diagram. The behavioral parts of service contracts are also derivable if one follow the messages and flows in the BPMN diagram focusing on the service lanes, in addition to how and who the service are communicating with. A good correlation between capability's and needs, which gives a score of a good 1 almost 2.

66 REQ6: CIM2PIM Transformations

The transformations in POSI are focusing on inter CIM model mappings and does not elaborate on transformations between CIM to PIM. However the structure conforms well with the SoaML service architecture model described in figure 35.

67 REQ7: Process support

The process support in POSI is more or less a mix of OASIS and COMET-S. In addition there is added the different phases that is mostly coherent with the perspectives in part 54. 2 points

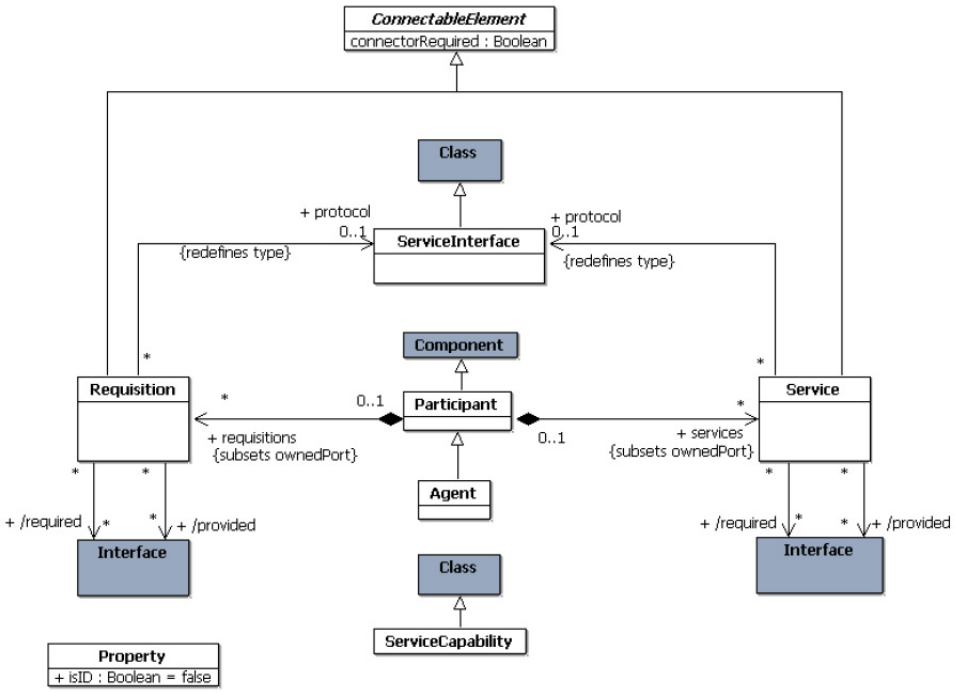


Figure 35: SoaML profile

68 REQ8: Tool support

There is no speciffic tool that supports the whole POSI Methodology. The need for a colaboration tool could be covered by a simple wiki, but organizations that use ARIS can use the build in publishing enginethat export diagrams and documents to html. The tools to support the modeling of the architecture can be done for instance on the EclipseIDE editor or with ARIS. POSI has no fully integrated toolsupport, which gives 1 point.

69 REQ9: Holistic view

POSI provides a way of describing the services anchored in business goals or motivation. It also has a process, language and notation that has a holistic perspective describing the SOA. However the transformation down to PIM level is uncomplete and this means 1 point on this requirement.

70 POSI summary table

This table sumarizes the points for POSI.

Part XI
Conclusion and further Work

The conclusion and further work part is first comparing the results from the analysis of the methodologies and then highlighting weaknesses and problem areas that need to be resolved. In this paper it has been investigated how the different methodologies are approaching service identification in a service oriented architecture. The results of the reseach has shown that when looking at architecture from a conceptual view different aspects are arising. The aspects of information, rules , organization, process, services and non functional. This gave a pointer of how to set up requirements and validating that the methodology has taken all aspects of the service architecture in cocideration. Here is a summary of questions that are raised in the thesis. All these should be adressed in the conclusion.
1.There are different tecniques used to identify services in a SOA, which?
2.How to identify core service attributes.
3.How to do an analysis of a SOA methodology.
4.Can services be categorized in perspectives?

Table 6: POSI summary table

Requirement	Key elements	Value
Service identification techniques	Bottom up Process Capability Collaboration Use case Business Service Architectural aspect	2 2 2 2 2 2 2
Service attributes	Use case,Need capability ,BPMN	2
Notation and language	OASIS need,UML use case, BPMN and SoaML	2
Information model	UML-class diagram and OWL	2
Service description	Use case, BPMN and SOA-Pro	1
Transformations	Partly described	1
Process	POSI service lifecycle, roles and workshops	2
Tool support	Some eclipse tools, RSM	1
Holistic view	Aspects, layers, tools, process	1

5.Can these techniques found in no 3, be composed in a certain order, to identify services with respect to the different service perspectives?
6.Does POSI add something to the SOA methodology landscape?

71 Conclusion

The conclusion is done by summing up the results from the analysis and then describe what the results are expressing according to the questions raised in the thesis.

71.1 Evaluation summary

In the evaluation summary table it is obvious that POSI has a high score because it is using all the techniques described in the requirements part. The compositions of the tecniques used may not be the optimal configuration for all possible cases, but POSI is definently decribing a methodology that collects all the information expected. Thous it is not giving detailed descriptions of how to transform the service models but only providing a conceptual mapping down to a PIM level model. The product value based technique could be replacing the need capability tecnique, because the interaction services are intuitively right for seeing services from the end customer and external partners that are offering one or more services as a product providing a value that is needed in the business quest for creating profitt. A service is required to create something more than the sum of all added components. If not it may never be concidered a value needed by the participants and still giving value to the business.

Requirement	**OASIS**	**COMET-S**	**ARIS**	**ArchiMate**	**POSI**
1.1 Bottom up	0	0	2	0	2
1.2 Process	0	0	2	0	2
1.3 Capability	2	2	0	2	2
1.4 Collaboration	0	2	0	1	2
1.5 Use case	0	2	0	0	2
1.6 Business Service	0	0	0	1	2
1.7 Architectural aspect	1	1	0	1	2
2. Service attributes	2	1	2	2	2
3. Notation and language	1	2	1	2	2
4. Information model	2	2	2	0	2
5. Service description	2	1	2	1	1
6. Transformations	0	1	0	1	1
7. Process	2	1	2	2	2
8. Tool support	0	1	0	0	1
9. Holistic view	1	1	1	0	1
SUM	13	17	14	13	25

Table 7: Evaluation summary table

71.2 Service identification

Are there many Service identification tecniques, and are they fit to identify different service aspects? As seen in the evaluation summary table 7 there where identified 7 groups of service identification tecniques used in the evaluation of the methodologies. These teqniques are described in section 13. The thought was that each of the techniques actually was representing and fitting with the different ways to identify different service perspectives. It would be wrong to say that there is an one to one relationship between the tecniques and the service perspectives, but there are some coherence. Especially with the process driven approach and the process services. Also the need capability technique fits well with the idea of fitting the SOA capabilitys to customer need and choosing partners by fitting their capabilities to the business needs. Having a methodology that covers different service identification tecniques enables a flexible way of identifying services and fitting the methodology to the organization. When working with all these concepts and tecnniques, it arises a question? Is the configuration of the different service identification techniques optimal, and is there any issues that may alter the optimal configuration of the sequence of tecniques. It is hard to say if by using all these service identification tecniques, that the service architecture gets better or more complete with respect to the quality or quantity of the services identified. However one may say that the services identified are based and anchored in the organization's strategy and position. What discovered is that the tecniques can be combined and used in different strategies, competitive and open or protective and hidden. Initially i would propose a two axis model seen in figure 36. vertical axis: open-closed horizional axis: competitive-protected. Competitive organizations are typically a sales organization that has a focus on filling the need of some customer. But the need of a customer or a participating stakeholder on background of some market analysis, may not be the focus of the organization. In a protected organization like a goverment where the focus is not to serve the customer but to secure the integrety and the quality of the work beeing done. The organization may also be an information intensive organization that sells information or processes information on demand. The main goals of this organization is to increase the ability to enable comunication and collaboration with other organizations and then you may base your top services on collaboration driven service identification keeping an transparent open view.

In a perfect world (see figure 36) , an organization may have a clear position, but in the real world it may be hetrogenous. Meaning it may be open or competitive in one context and protected and closed in another context. The point with these issues is of course that the service identification tecniques used posesses distictive qualities that makes them better fitted in one context that the other. How the tecniques are best configured to fit different contexts

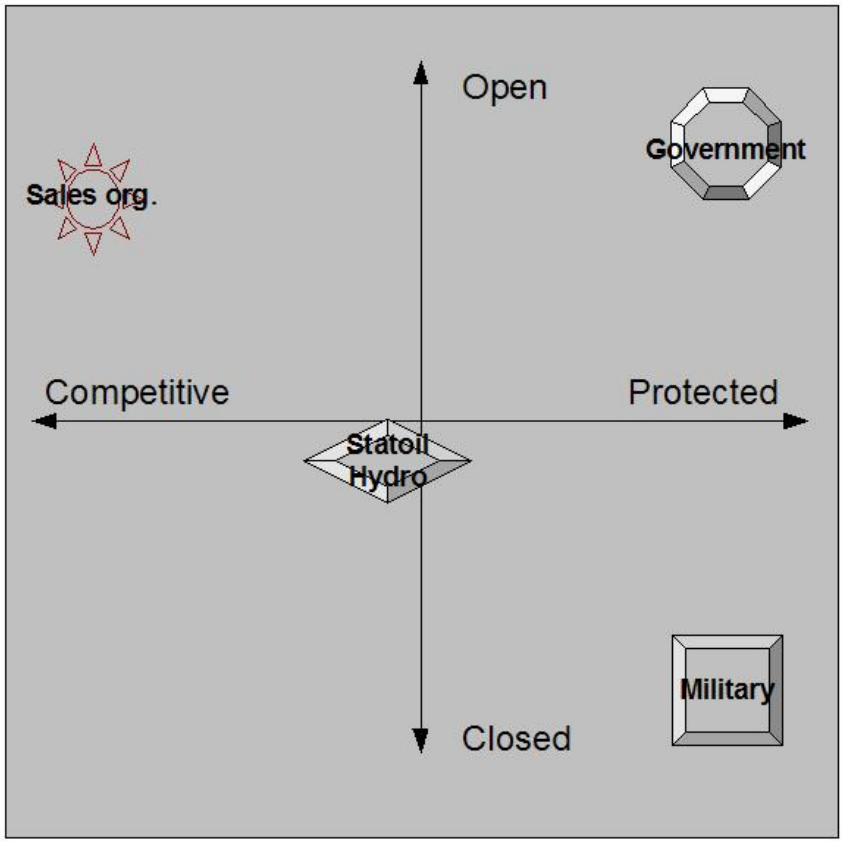


Figure 36: Organization position

may be explored in further work.

71.3 Service attributes

How to identify core service attributes? This question is answered by the evaluation of requirement 2 in section 13.2 where results for POSI in the evaluation summary table shows that this requirement is achieved.

71.4 SOA methodology analysis

How to do an analysis of a SOA methodology? To evaluate the methodologies a set of requirements has been established and a quantification has been defined and structured and applied. According to the requirements, an analysis of state of the art technologies has been designed and executed.

71.5 Service categorization

How can Services be categorized in a way that is suited to be used as a support for the design and management of the service oriented architecture? The perspectives as they are called in this contex are described in section 54. The strongest argument for beleiving that POSI has given an answer to this question is that the categories is deviding the concerns and enables reusability. It also enables for instance the design implementation and administration of security, authorization issues to focus on the interaction services.

71.5.1 Does POSI add something to the SOA methodology landscape?

Looking at the result POSI has the potential to represent more than the components added initially by the state of the art technologies. POSI tries to merge the results of the analysis into a way of thinking service identification and design. This means that SOA developers do have the freedom to fit the focus of the service identification to the organizations inherent properties. There has been established a way of seeing the organizations position seen in figure 36. There may of course be other concepts that can pinpoint the organization's position in relation to which service identification tecniques to use. Then assigning identification techniques to each of the service perspectives. Some of the tecniques have an obvious link to a perspective, like the bottom up service identification technique that is good for identifying services to reuse or the process driven service identification. The usage of the need capability driven technique is not quite as obvious. The methodology does not either favorise any vendor and the methodology can be fit to the resources that allready exists in the organization.

72 Quo Vadis

There are issues that needs to be resolved. Questions like what are the inherent properties of these service identification tecniques? This subject has been looked upon in this paper. There has been identified some different service identification tecniques, but there may be some that are not represented and also their properties needs to be explored and documented further. Another issue that arises is how the organization can be positioned to create a foundation for deciding what techniques to use. Is there other concepts to use in the organsation position graph that are more descriptive or fitted to position the organization in relation to the tecniques? The perspectives that are a part of POSI and not described and elaborated in this paper should also be explored and described to complete the methodology. An other issue that sould be done is to conduct a thorough evaluation and validation of POSI. POSI has been applied to the buyer seller example used in SHAPE [32] only partly, but a more comprehensive examplicification may be required to validate POSI. The tool support for POSI is not very good. Further work should also involve defining and developing better tool support, which in its turn makes it easyer to conduct execution and validation of POSI. But its important that POSI stays vendor neutral to remain flexible.

References

[1] M. Aanestad, The camera as an actor: Design-in-use of telemedicine infrastructure in surgery, 2003, Computer Supported Cooperative Work: The Journal of Collaborative Computing, vol 12, pp 1-20.

[2] Zimmermann, Elements of service-oriented analysis and design, June 2004, http://www.ibm.com/developerworks/library/ws-soad1/.

[3] J. Klueckmann, 10 steps to business driven soa, on web, 2007, http://www.ids-scheer.com/en/ARIS/ARIS_Software/ARIS_Expert_Paper_Library/.

[4] B. et al., Inf5120-part-iii-mde4soa-comets, http://www.uio.no/studier/emner/matnat/ifi/INF5120/v08/undervisningsmateriale/.

[5] S. JONES, A methodology for service architecture, on web, 26th October 2005, http://OASIS.open.org.

[6] OASIS, Reference model for service oriented architecture 1.0, on web, 2 August 2006, http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=soa-rm.

[7] Oasis, Reference architecture for service oriented architecture version 1.0, on web, 23 April 2008, http://docs.oasis-open.org/soa-rm/soa-ra/v1.0/soa-ra-pr-01.html.

[8] Zachman, A framework for information systems architecture, 1987.

[9] A. J.Berre, Architectural aspect matrix, 2008, http://www.shape-project.eu/.

[10] T. Earl, *SOA design patterns* (Prentice hall, 2008).

[11] ThesesusTexoConsortium, "the inter-enterprise service engineering", accessed juni 2008, http://theseus-programm.de/scenarios/en/texo.

[12] Gos, About theserus, 2006, http://theseus-programm.de/about-theseus.

[13] Arsajani, Service oriented modelling and architecture, 2004, www.128.ibm.com/developersworks/webservices/library/ws-soa-design1/.

[14] F. A. Cummins, Building the agile enterprise with soa, bpm and mbm, 2008.

[15] Hotle and Blechar, "service oriented development of applications", 2004, accessed nov 2008, http://www.gartner.com/resources/123500/123523/soda_reuse_and_.pdf.

[16] Allen, Sae- soa adoption and excellence, 2007, http://www.cbdiforum.com/secure/interact/2007-02/service_oriented_processes.php.

[17] Butler, The architecture component of the sae reference framework for soa, March 2007, http://www.cbdiforum.com/secure/interact/2007-02/the_architecture_component.php.

[18] L. et Al, Smart- service oriented migration and reuse tecniques, 2005, www.sei.cmu.edu/pub/documents/05.reports/pdf/05tn029.pdf.

[19] Lefever, Service centric system engineering, 2005, accessed Dec 2008, http://www.secse-project.eu/wp-content/uploads/2007/08/a5_d4-secse-methodology-v1_3.pdf.

[20] the OPEN group, Archimate website, 2008, http://www.archimate.org/en/about_archimate/.

[21] M. Lankhorst, Enterprise architecture at work, 2005.

[22] A.J.Berre, Comet component and model-based development methodology, 2007, http://modelbased.net/comet.

[23] SINTEF++, Athena interoperability framework, 2007, Athena project website http://www.modelbased.net/aif/index.html.

[24] SINTEF++, Service-oriented development in a unified framework, 2007, Sodium project website http://www.atc.gr/sodium/project.asp.

[25] SINTEF++, Swing, 2007, Swing project website http://www.swing-project.org/index.html.

[26] R. Davies, Business process modelling with aris, 2001.

[27] OMG, Service oriented architecture modeling language (soaml) - specification for the uml profile and metamodel for services (upms), October 2008.

[28] W3C, Ontology web language, 2004.

[29] T. O. Group, Togaf, 2002, accessed juni 2008, http://www.opengroup.org/architecture/togaf8-doc/arch/site-map.html.

[30] O. G. Consortium, Iso19119,geographic information, 2001, site accessed juni 2008, http://www.isotc211.org.

[31] I. Workshop, Thirteenth ieee international workshops on enabling technologies: Infrastructure for collaborative enterprises, June 2004, IEEE Workshop on collaborative Enterprises proceedings.

[32] A. J.Berre, Semanticly enabled hetrogenous service architecture and plattforms engineering, 2008, http://www.shape-project.eu/.

Part XII
ARIS SOA

A ARIS

ARIS is an acronym for ARchitecture of integrated Information Systems. ARIS is mainly a concept or a methodology that was developed by August-Wilhelm Scheer at the institute fur wirtschaftsinformatik at the Universitat des Saarlandes in Saarbrucken, Germany. It is also a computer based tool that enables you to model businesses on the ground of its methods. The methodology aims to close the gap between business theory and information and communication technology. This means expressing concepts of your business in such a precise way that it may be analyzed in detail and used as a baseline for the development of information systems. The ARIS tool represents the business by process models, systems, organizations, software, costs, data and so on. At last but equally important, is ARIS'es ability to model the relationships between them.

ARIS has several solutions for modelling and analyzing businesses. In addition to the standard Business architect/designer ARIS also has a SOA architect which integrates service description diagrams with the Business process diagrams. They do not take any position in the question of starting with services and building business processes around it, or to start with process and then integrate a service architecture on top. The important thing they argue is

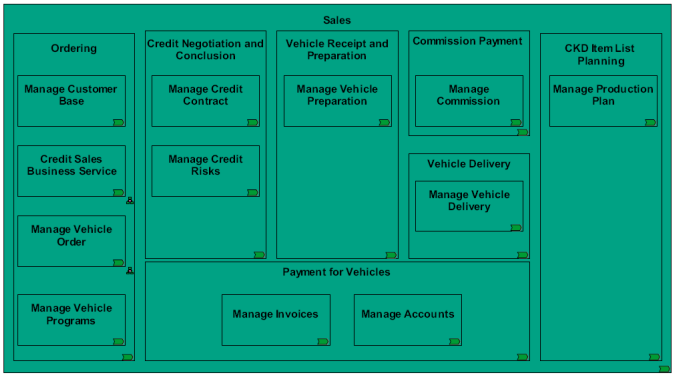


Figure 37: Service breakdown

that there is a service architecture present integrated with the enterprise architecture.

A.1 Service descriptions

ARIS business oriented service description corresponds to the CIM(computer independent model) level, the highest level of the service. This level contains no technical details and offers a top-down view of services. It describes what the service offers, not how it is implemented. The lower levels is a question about how it is implemented.

The different service diagrams:

The first step in describing the services is the service architecture diagram. This first diagram is the breakdown of sales services.

The high level services is then further broke down into the service and its capability's. Below we can see how it is expressed with another service architecture diagram.

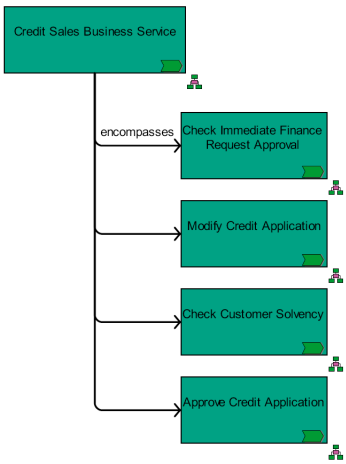


Figure 38: Service capability diagram

To describe the services in more detail a service assignment diagram is used. This diagram is however a variant of the function allocation diagram. This would enable us to assign input and output data structures, actors and capability's to the service description.

The last of the service diagrams is the access diagram that creates a link between the software service and the web service. The software service specified as the realization of the business service is still a platform independent representation, because there are no technology specific information in the diagram.

When you have completed the service diagrams, they should be connected to the business processes. You can integrate it with an existing business model or there is the possibility to design your processes around the services to reach for the "true SOA" experience.

A.2 Business Model

The business model is the way end-to-end processes is nested together for the purpose of achieving the businesses goals. In a business model one must be allowed to express the business logic of a specific firm. The business model

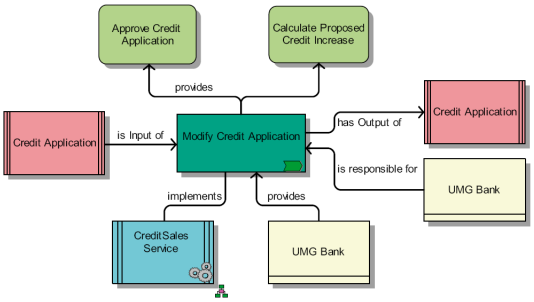


Figure 39: Assignment diagram

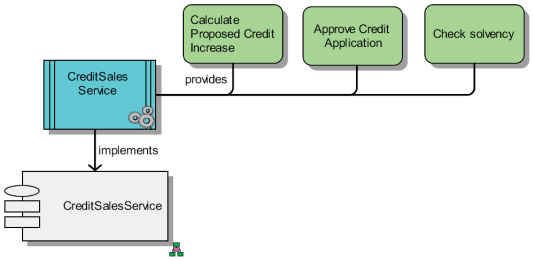


Figure 40: Service Access diagram

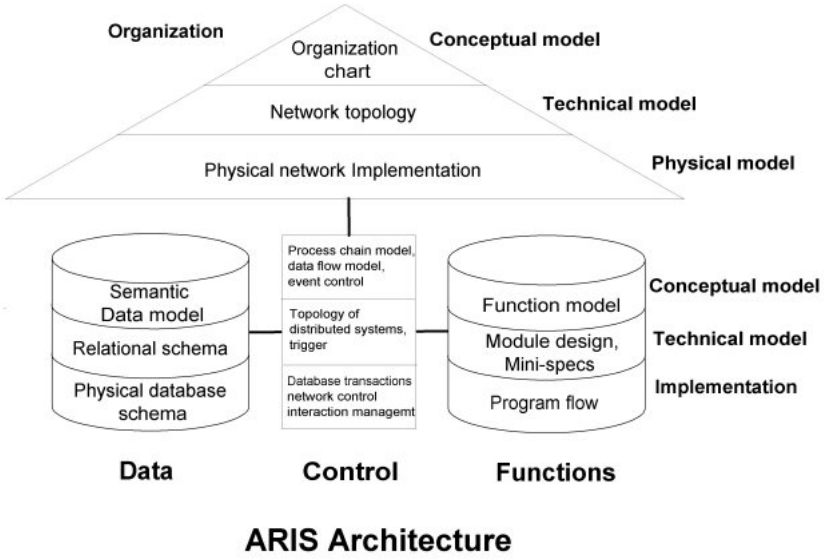


Figure 41: ARIS architecture

should describe the different aspects of your business and its relations. The ARIS tool or method if you like, defines these aspects as organizations, control, data, and its respective functions. In addition it should say something about the output products and services. The ARIS house is the representation of the concepts or views used to model the business model, which also reflected in the differing parts of the meta model.

A.3 Business Process

Business Process in ARIS context is the definitions of the tasks and the sequence of those task necessary to deliver a business function[3]. Business processes describe how a business pursues its objectives.

> "If it doesn't make 3 people angry it isn't a process[26]"

A process is either adding value to the business or fulfilling some necessary function. The process is a transformation witch takes input and generates the output as a service or a product. Processes are pieces of a big puzzle that is the Business model. In other words an ordered sequence of business activities witch is modelled at different abstraction levels. A process takes something as a input, transforms it into the output.

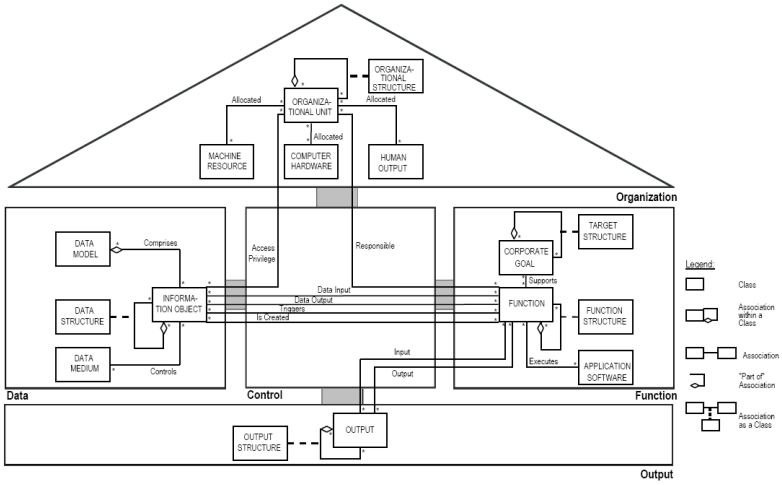


Figure 42: ARIS meta model

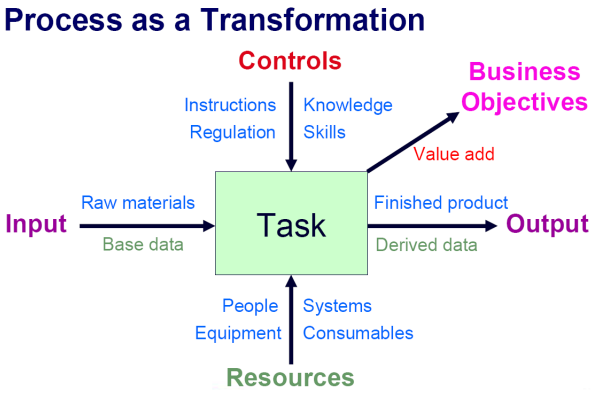


Figure 43: Process as transformation

A.4 Business process modelling

ARIS is a process modelling centered method. Process modelling is the documentation, capturing, analysis and design of structure of business processes, their relationships with the resources needed to implement them, and the environment they will be applied. The ARIS method is strongly inspired by the Zachman framework. This is of course reflected in its meta model that is composed of the five parts mentioned earlier namely organization, control, data, function and output. But this is the concepts used to represent the business model, before trying to map out the processes one should investigate some areas of special interest for what processes to model. These are the businesses motivation and purpose for existing. These are the What, How and the Who:

- Value proposition. What is sold and delivered to the marked
- Supply chain. How is it created and delivered to the marked
- Target customer. To whom is it delivered.

The highest level or the top down overview is the process map. The process map is a "added value chain diagram" and could look something like this. It purpose is to create the big picture of the business, and it will serve as a roadmap for the further refinement and process modelling.

The process map (figure 44)can include some technical terms, major process groups and their categorization and to some extend also some of the subprocess. This charts provides us with a high level view of the business segments, and is useful to create a vision of our business model. It will help us to pinpoint what is core and support processes and their location in the businesses landscape. ARIS provides several generic templates for domain specific areas.

To refine the picture the organization is modelled in a organizational chart with departments, positions, roles and resources like information systems and applications. These objects are to be reused in the EPC diagrams and FAD diagrams.

Next level is process models flowcharts, which is extended with enough information or semantics so that the process can be analyzed, simulated andor executed. At this level of modelling the event driven process chain(EPC) is the most used diagram, see figure 45. However ARIS also supports BPMN. However this level of modelling may contain several layers of EPC charts that

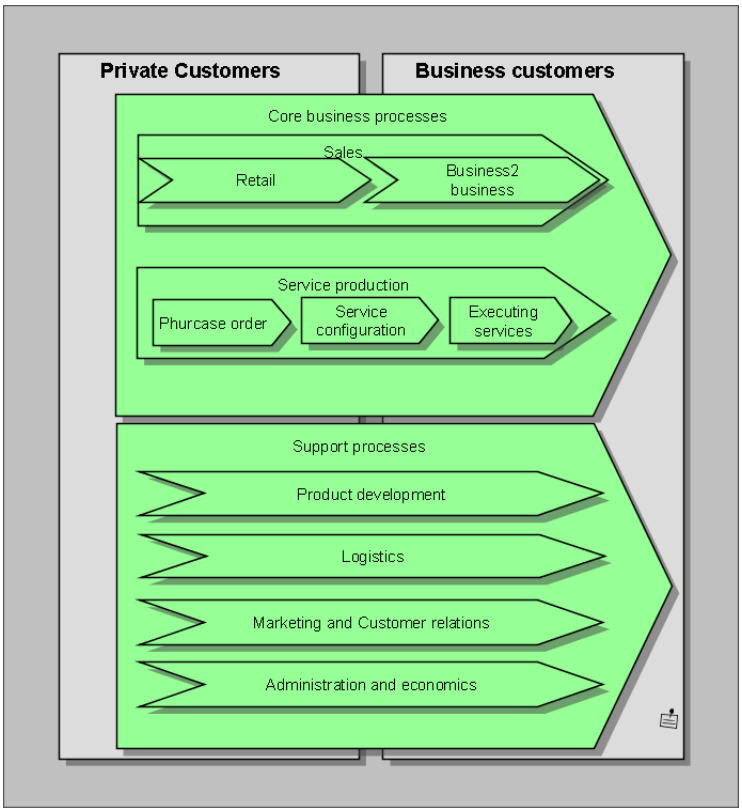


Figure 44: Example of a Process map

are woven into each other. This means that a function can be extended into new EPC flowcharts if it does not represent the process at a sufficient detailed level.

The next level is for specifying a functions input, output, relevant actors and data types involved. For this purpose the function allocation diagram(FAD) is widely used. The ARIS tool provide transformations from EPC or BPM, to BPEL or WSDL. To take advantage of this structures one may load them into a Service Oriented Architecture(SOA) which will execute the business model. Seen in a holistic view it all can take part in a Business Process Architecture.

The modules in the ARIS platform provide means for a computer-aided analysis, planning and introduction of managerial information systems. http://www.ibm.com/developerworks/library/ws-soad1/

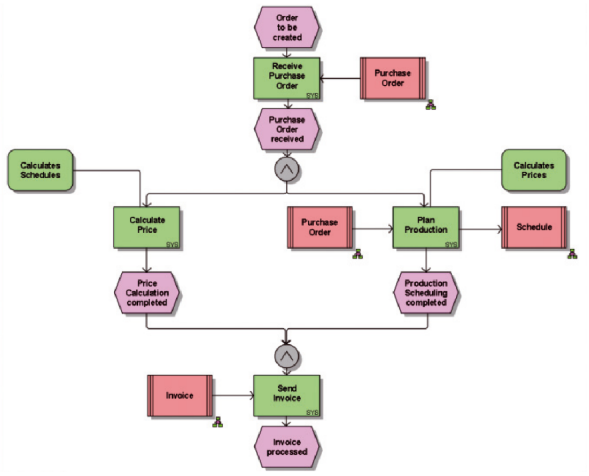


Figure 45: EPC diagram example

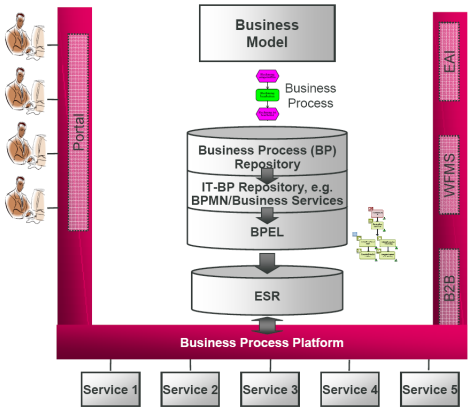


Figure 46: Business process architecture

Part XIII
COMET-S

B Comet-S

B.1 Introduction

Comet-s is a methodology which is promoting a model based methodology in a three layer architecture, Business model(CIM), Requirements model(PIM) and the Service architecture model. The starting point for the COMET-S methodology (COMET for Services) is the existing COMET methodology as documented at www.modelbased.net/comet. As you can see from the figure below 47 the focus for Comet-s is the Service, Process and Information areas.

Comet-S also provide some guidelines for the development process and notation of models. The processes and techniques introduced her is based on the COMET methodology developed mainly by the related projects ATHENA[23], SODIUM [24] and SWING [25]. The information about the Comet-s methodology is taken from the MDE for SOA [4]. The COMET-S methodology is compiled of four different modelling areas, business- model, Requirements model, Architecture model and platform specific model. It is using the newly available meta models from the OMG standardization projects. In particular for the CIM level BMM and BPMN are proposed, and for PIM the UPMS. The figure below gives an overview of the four main modelling areas of the Comet-S methodology.

- Business model
- Requirements model
- Architecture model
- Platform specific model

B.2 Business model

The business modelling is used to outline and describe the part or role played by the product being developed. In addition this will be linked to the product's context that is driving its development. The modelling of a business is a

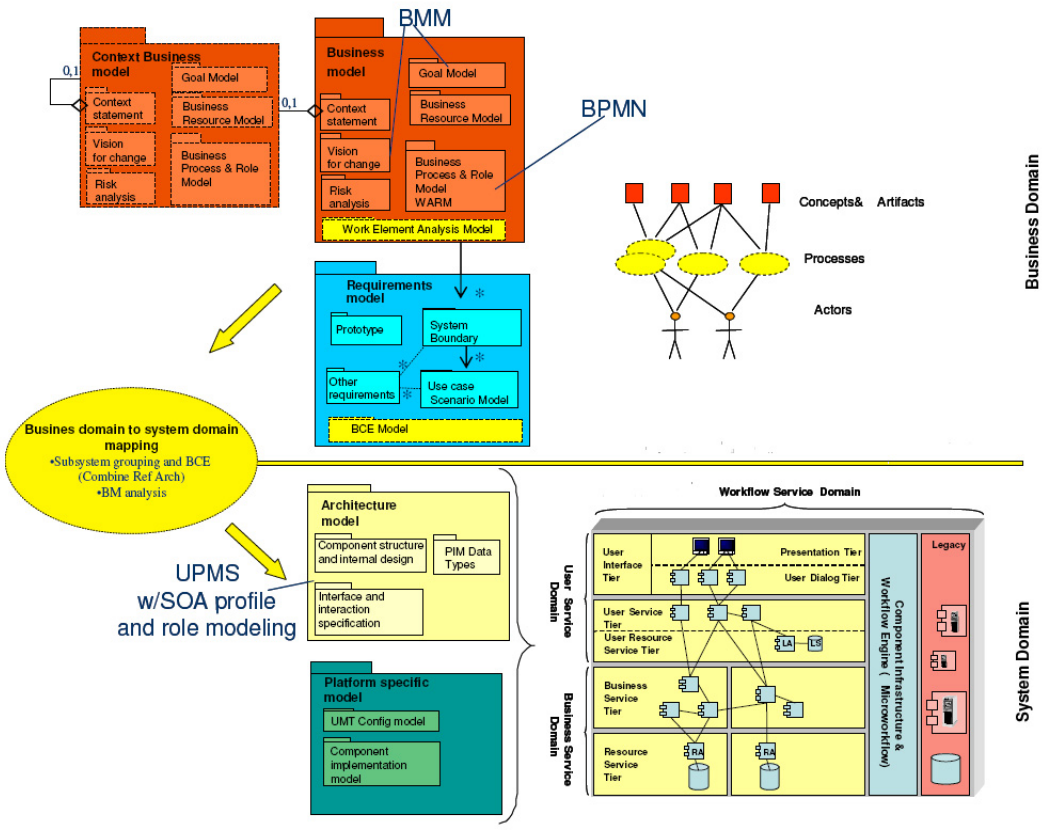


Figure 47: Three layer architecture

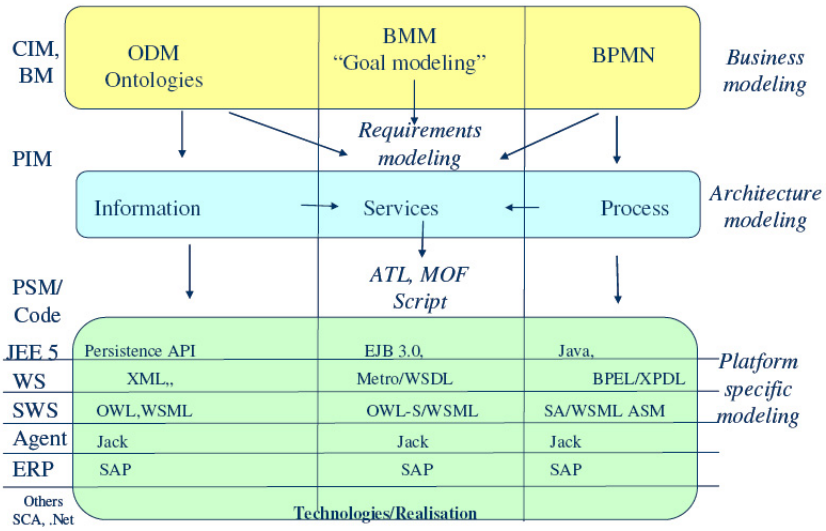


Figure 48: Modelling areas

recursive activity and some of the models could easily become multi-layered. Meaning a process can contain subprocesses that is defined in separate models.

Process:
In the work of collecting domain information, all with interests in the product should attend and contribute in these activity's. The possible Business stakeholders involved:

- Decision makers for authorization of funding for the development of the product.
- People who are responsible for design and maintenance of the business processes to be supported by the Product.
- Product consumers
- Decision makers for acceptance of the Product
- Managers of operation for the product

Notation and language:
The business supported by a service architecture have goals as one of the primary drivers for what services to define and evolve. Other primary drivers for services is processes required to meet goals and the roles to fulfill them. The community structures groups of resources with a common or interleaving goals. The business model consists of these work products:

- Scoping statements including context statements, vision for change and risk analysis
- Goal model describing the business goals being realized through developing, implementing and using the product.
- Community model including business processes, role modelling and business resources.

B.2.1 Scoping Statements with BMM

The Context statement is modelled by BMM concepts and including definition of the scope of the business model and its position in its context. This will give an 20.000 feet domain picture giving an overall understanding of the domain, identifying stakeholders, their relationships and concerns. The Vision statement is highlighting why this product should be funded, and giving a clear understanding of the gap between the current situation and where to go. The Risk analysis model is describing elements that might have an influence on the product, both good or bad. A return on investment(ROI) estimate should also be a part of this work product. Goal model. Goals of the major stakeholders is agreed upon and serves as a reference throughout the development process. The Goal model has a central role in visualizing high level business processes that will be used as a base for further analysis in the Business Process model. Community model. The community model is a container with a set of communities that are collections of resources working together in one or more processes to achieve one or more goals. Communities are essential for performing recursive, parallel and decomposition of both structure and behavior in business process modelling. The business resource model describes an information model identifying and defining relevant concepts of the domain(what it is) and the processes(what it does)that seeks to realize the goals from the goal model.

B.2.2 BPMN Business process and roles model

The Business process model is defining the processes in the domain that is needed to realize stakeholders goals or relevant to the product. In addition it will describe roles that performs the processes. As already mentioned the model may be at many levels, from the high level business processes down to the Work Analysis Refinement Model(WARM). The process model is derived from the Goal Model. Starting with the identification of the enabling behavior for the goals to be achieved. The different behavior's derived is then consolidated to a set of behavior's that covers all the goals. These processes is expressed through the BPMN language which has four distinct categories of concept elements.

- Flow Objects: Events, Activities, Gateways

- Connecting Objects: Sequence Flow, Message Flow, Association
- Swim lanes: Pool, Lane
- Artifacts: Data Objects, Group, Annotations

B.2.3 Business resource model

The business resource model is an information model of the main things and concepts relevant to the product and its context. The resource model is to be realized with UML class model or with a ontology model like OWL or a combination of both.

B.2.4 Work Analysis Refinement Model (WARM)

The WARM is a refinement of the core Business model with focus on work analysis, asking which kind of resources does the specific kinds of work. WARM is not a specific model but a refinement of the existing models, defining what kinds of steps are performed by whom, human or system?

B.3 Requirements model

The requirements model is identifying the system requirements including the functional requirements, non-functional requirements and constraints. The requirements model includes several sub models like the Use Case Model, a prototype, Non-Functional requirements and the BCE model. The Use Case Model consists of a System Boundary model and the Use Case Scenario model. The system Boundary model describes the System Boundaries, the actors and their responsibilities, and the services offered by the system. The Use Case Scenario model details the identified use cases. The prototype is made to reduce technical risks and ensuring user participation. It also contributes to the quality of user interfaces by testing critical use cases and other risky parts of the architecture. The Non-functional requirements describes requirements like efficiency, response , integrity and so on.

B.4 Business domain to system domain mapping

The BCE model provides the link between the Requirements model and the Architecture model. The model is the output of a technique used to extract the system domain models (architecture model and platforms specific model) from the business domain models (the Business model and the Requirements model). Requirements model to architecture model transformation are also mentioned in the COMET-S methodology. The following transformations are proposed:

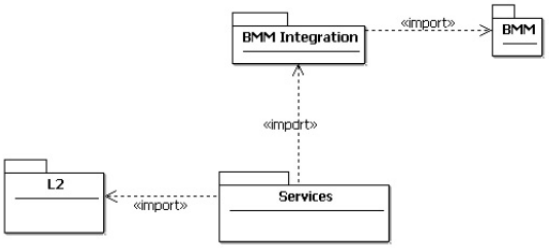


Figure 49: SOApro overview

- All actors is mapped to a UserService and there is created an Interface as a provided interface of the actor. For all the use cases that the actor relates to, add a corresponding operation to the Interface. Naming conventions: The actor name is used to name the UserService and the same for the interface only prefixed with an I. The operations will be named according to the use cases.
- Map each <<Manage>> use case to a BusinessService providing CRUD operations as well as find and collection operations for the Resource(s) that are related to the <<Manage>> use case. Naming conventions: use the resource name prefixed with Manage.

Include and extend relationships can be handled like this:

- Include -¿ reusable UserService with interface providing the include service or an operation in the interface of the extended UserService.
- Extend -¿ Operation in the interface of the extended UserService

B.5 Service Architecture Model

The Comet-S methodology is embracing the standards from OMG(Object Management Group), and are using the emerging UPMS standard as a framework for the service architecture. An open source implementation of the UPMS standard is in development by the European SHAPE IST project. Input to the standardization work has been provided also by earlier European projects like ATHENA, SWING and SODIUM.

The figure above 49 shows the overview of SOA-Pro meta model. The Services package is merged with the UML2 to extend the capability of service modelling. There are several other Integration packages that extend Services with other OMG specifications like the BMM standard and others.

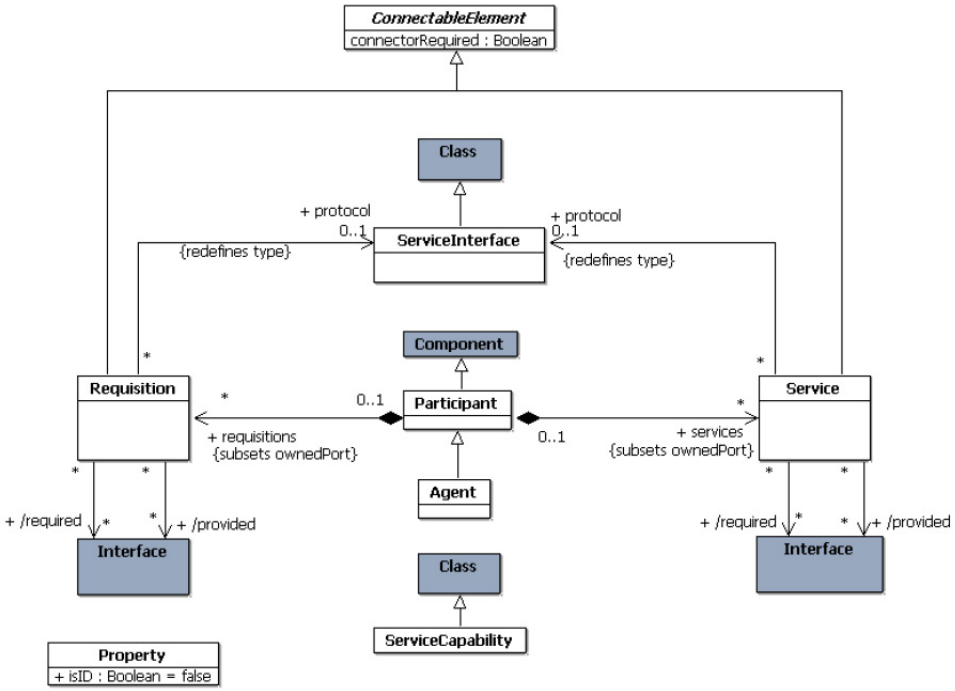


Figure 50: SoaML profile

B.5.1 Key concepts of service

One of the core elements are the concept of Service. The Service is a capability offered by one entity or entity's to others through well defined terms and conditions regulated by service contracts. These entity's are people, organizations or systems we call participants. 50

Participants offer or consume services through ports with the <<Service>> stereotype. The Service port is the interaction point between participants and have a type that describes how to use the service and may either be a UML interface, or a Service Interface like in 51. The interfaces are standard UML interfaces provided or required by the service interface.

The Service interface and and its enclosed parts specifies the roles that will be played by the parties involved with the service. The behavior specifies the interactions between provider and consumer, the contract of interaction illustrated by a interaction diagram or any other UML behavior specification.

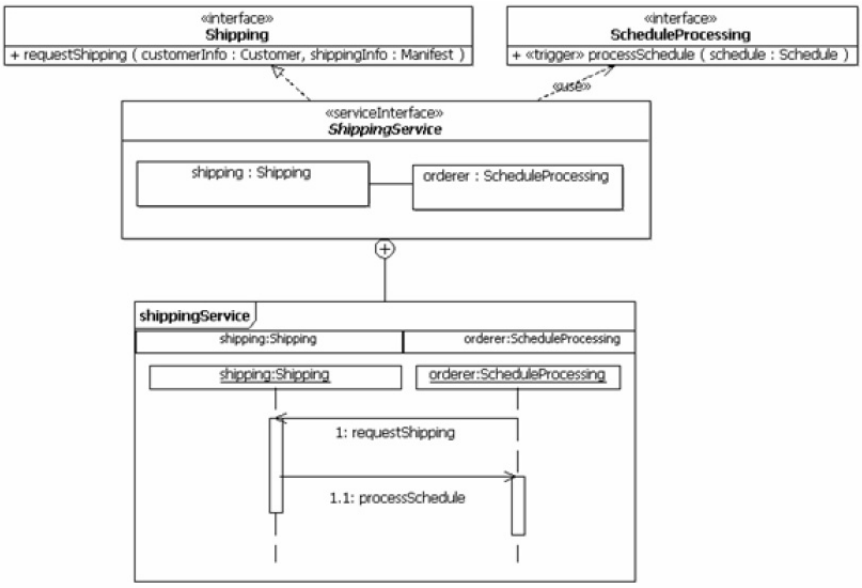


Figure 51: fulfilling service contract

B.5.2 Key concepts of Service architecture

The Service Architecture is a network of roles providing and consuming services to fulfill a purpose. The service architecture may be expressed as a Community architecture. The community architecture is describing multiple participants roles filled with participants or service realizations. In addition it describes their regulating service contracts that the interacting participants must agree to for the service to be initiated. Service contract- the service contract is describing the full specification of a service which includes all the information, choreography and any other terms and conditions of the service. 52. The UML collaboration is the basis for realization by a set of components. These can be described using UML2.0 composite structure models. The next steps is to provide a realization for the specification and each of its parts. First the specification that shows how the services are organized into a service architecture.

Then each of the parts of the specification must be realized by a part with the same name and a compatible type in the realizing participant. In 53 it is shown as a composite structure, where the stereotypes <<service>> and <<requisition>> on ports are used to denote the required and provided services. Alternatively one may use the conjugate type sign as shown for the Invoicing Service.

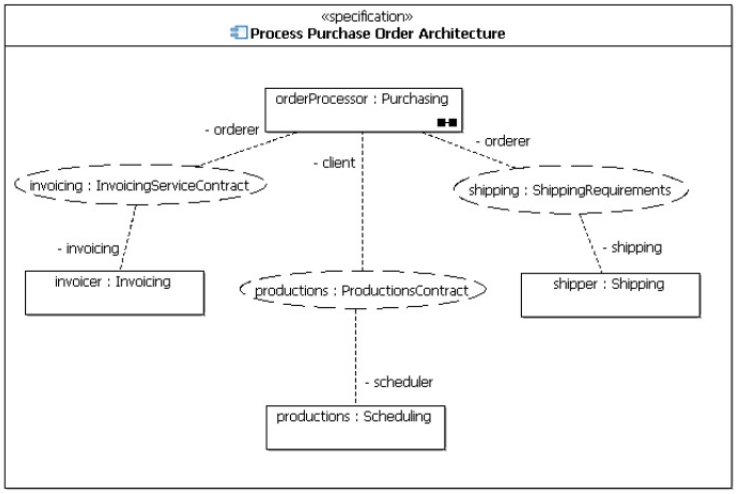


Figure 52: Service specification

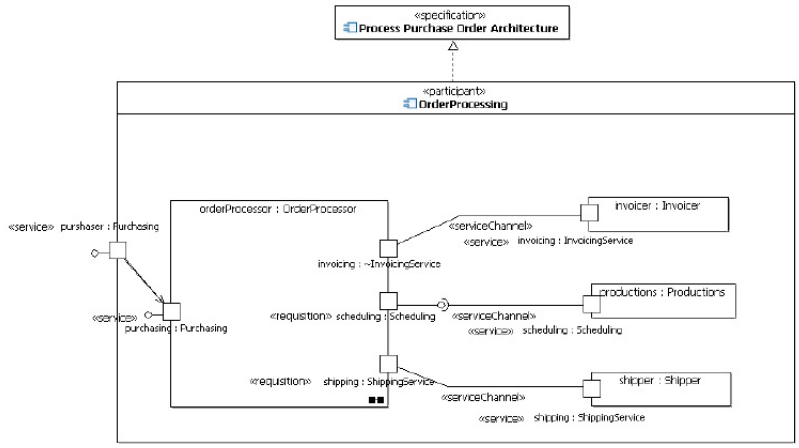


Figure 53: Realizing participant

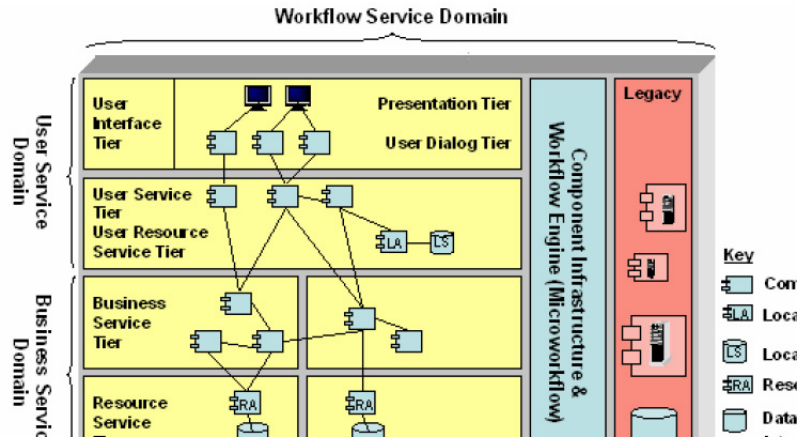


Figure 54: 4 tiers

The service reference architecture defines a set of logical tiers, that consists of a set of components. As you see in 54 it is divided into the user service domain and Business Service Domain with its respective tiers.

B.6 The PSM model

The Service architecture model is to be transformed into a platform specific model which contains the:

- Platform Profile Model which specifies the system in alignment to the actual technology profile for the specific platform.
- Component Implementation Model, which describes the implementation of the component specifications in a given programming language like JEE

Part XIV
ArchiMate SOA

C ArchiMate

C.1 Introduction

Enterprise architecture is an important instrument to address this company-wide integration. It is a coherent whole of principles, methods and models that are used in the design and realization of the enterprises organizational structure, business processes, information systems, and IT infrastructure. A well defined Enterprise Architecture enables an organizations to align business processes and IT operations with its strategy. A organizations is a living and dynamic entity and the architecture must be flexible and able to respond quickly to changes and optimization requirements.

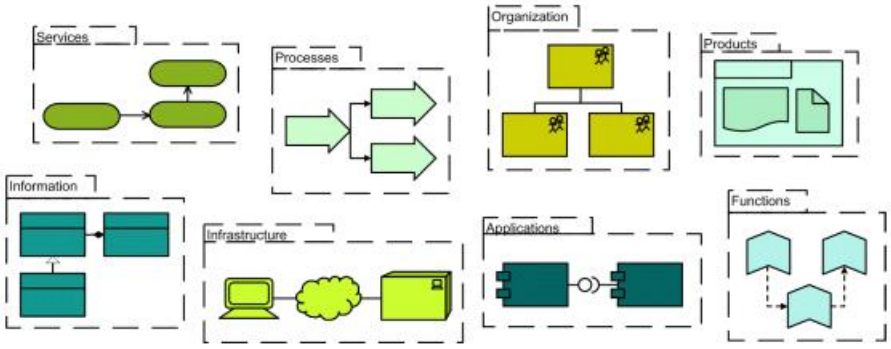


Figure 55: Archimate domains

Within companies various domain architectures can be found in Figure 55 , like organizations, business process, application, information, and technical architectures. Each of the various domains have been assigned their own set of concepts for modelling and visualization. The domains often overlap and use different notions to express the same ideas. Archimate is a tool and a modelling technique (language) that is trying to reach for a unified way of modelling enterprise architectures. Archimate seeks to integrate the various domains and describing them in a easily readable way. It has a clear set of concepts within and relationships between architectural domains. Archimate is of course not an isolated development. The relationships with existing

methods and techniques, like modelling languages such as UML and BPMN, and methods and frameworks like TOGAF and Zachman, are well-described. All information and illustrations in this Archimate summary is collected from the Enterprise at Work [21] and the archimate web site [20].

C.2 Service Orientation in Archimate

Services play a important role in the relationship between domains.[**?**] Service orientation supports developments such as the service-based network economy and ICT integration with Web services.

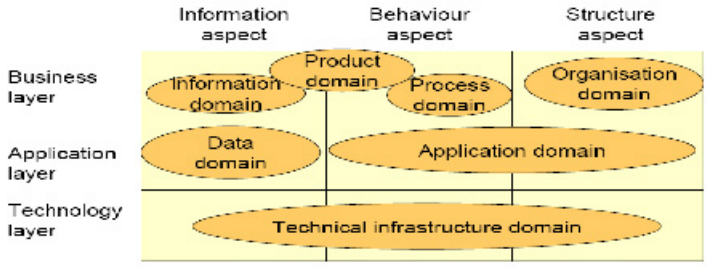


Figure 56: Archimate aspects and layers

The service is in this context defined as a unit of functionality that some entity makes available to its environment, and it provides some value for its service users.

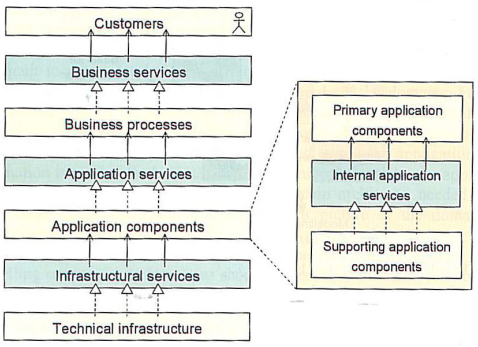


Figure 57: Archimate service layer stack

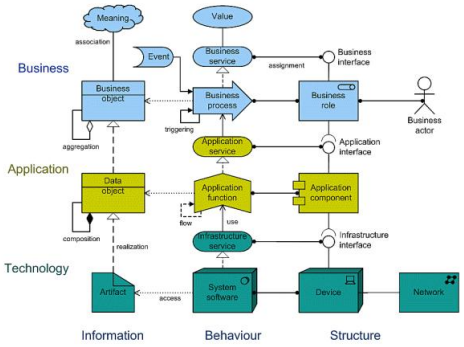


Figure 58: Archimate concept overview

Service orientation typically leads to a layered view of the enterprise architecture like in Figure 56 , where the concept of service is one of the main linking pins between the different layers. This will lead to a stack of service layers and implementation layers in Figure 57.

C.3 Service layers

Service layers with services made available to other layers are interleaved with implementation layers that realize the services. There might also be internal services, e.g. , services of supporting applications that are used by the end-user applications. For specific layers more concrete concepts are used.

The architecture are due to this divided into three distinct layers:

- The business layer which confirms to the CIM level. The business layer offers products and services to external customers, which are realized in the organizations by business processes performed by business actors or roles.
- The Application layer supports the business layer with application services which are realized by (software) application components. This layer is equal to the PIM level.
- The Technology layer offers infrastructural services (e.g., processing, storage and communication services) needed to run applications, realized by computer and communication hardware and system software. This layer is equal to the PSM level.

The most important concepts of Archimate are shown above 58. You can clearly see the uniform approach across layers. As one can observe from the

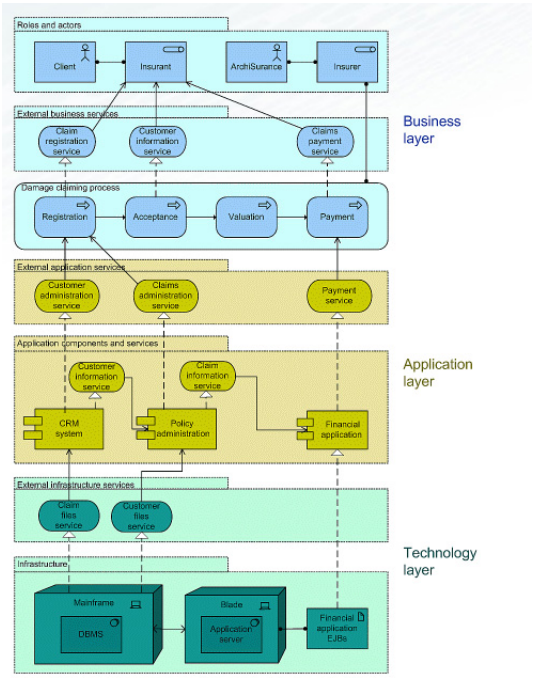


Figure 59: Archimate example model

figure above there is in addition to also a three row distinction in this matrix. First there is the passive structure or information, which is objects that are affected by the behavior. Typically this is information objects in the business layer and data objects in the application layer. Secondly the behavioral aspect. Behavioral concepts are assigned to structural concepts to show who or what displays the behavior. Thirdly is the active structures like roles, interfaces, collaborations, devices and components. In the example model below 59, you can see the integration from the technology layer, with e.g. the mainframe on which the database runs, via the application layer, with e.g. the policy administration, all the way up to the business layer, with the client who wants to register an insurance claim and the business process that provides the necessary services.

C.4 Business layer concepts

C.4.1 High level business concepts

Central high level business concepts makes it possible to describe and model the vision and business ideas.

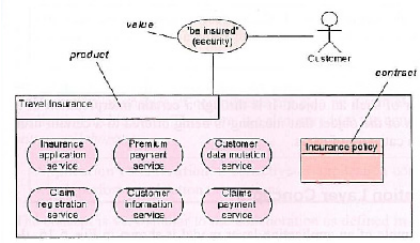


Figure 60: High level business concepts

Further on seeing them through service glasses and linking the operational side of the organizations to its business goals 60. Service orientation in this context is a service or a collection of services, that represents a product. The product that has some kind of value for a customer, is regulated by the service contract. The contract is a formal or informal specification of an agreement that specifies the characteristics, rights and requirements for the product and services use.

C.4.2 The business structure concepts

From the figure below 61 we can see how the main structure concepts are used. The central concept are the business actor that plays a business role in a business collaboration.

C.4.3 The behavioral aspect

The business roles use the business services which are realized by the business processes 61. The business interactions are linked to the structural business collaboration. Important concepts not represented in this illustration are the business activity and business function that are subclasses business behavior, like business processes.

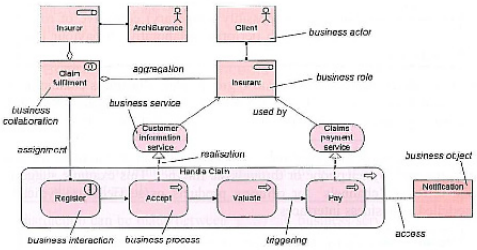


Figure 61: Business level concepts

C.4.4 The passive information aspect

The business information concepts are the representation of a concept that are implemented as business objects. In addition there is the notion of value, product and contract.

C.5 Application layer concepts

C.5.1 The application structure concepts

As we see in the concept overview (in figure 8) There are corresponding concepts to those in the business layer except for the actor and event in business layer. There is the application component, collaboration and interface.

C.5.2 The behavioral aspect

For the behavioral aspect there exist the concept of application service, application function and application interaction.

C.5.3 The passive information aspect

The information aspect is covered by data objects, which is the implementation of business objects.

C.6 Technology layer concepts

C.6.1 The technology structure concepts

Core concepts of the structural concepts are the infrastructure interface, node, communication path, network and devices. There are three classes of devices that is corresponding to the three behavioral infrastructure services. These are the

- computing devices
- storage devices
- networks

C.6.2 The technology behavioral aspect

For the behavior aspect the methodology provides the infrastructure service and system software. The infrastructure services are classified into three main types

- processing services
- data storage and access services
- communication services

C.6.3 The passive information aspect

This aspect is covered by the artifact concept which is the physical piece of information, used or produced in the development process, during deployment or by operating the system. Artifacts realizes data objects and application components. Below 62 a overview of the Archimate concepts and their notion 63.

References

Her kommer det fra shape dokumentet

Part XV
OASIS SOA works

D SOA with OASIS

The SOA adoption blueprints can be seen as a set of functional descriptions of a service identification process. It provides a business problem statement, a set of business requirements and a normative set of functions to be fulfilled where vendor specific details are abstracted. It is supporting the use of the OASIS

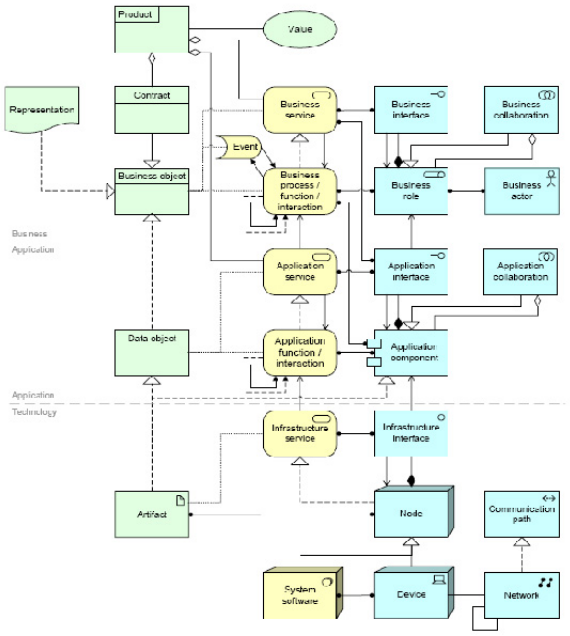


Figure 62: Archimate concepts

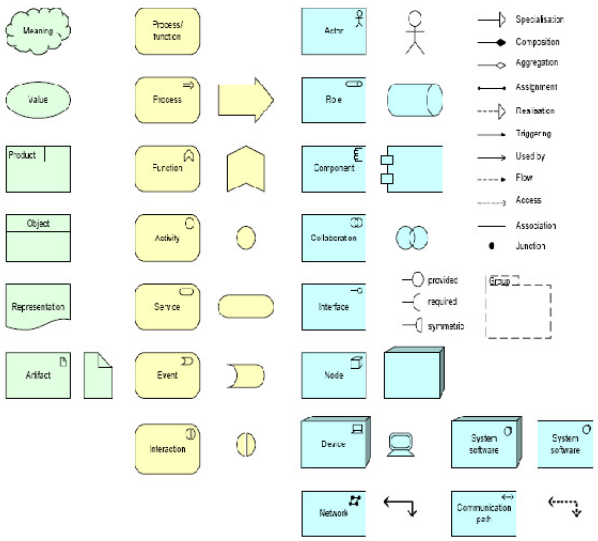


Figure 63: Archimate symbols

SOA reference model[6] and reference architecture[7], which spans over the whole Service oriented architecture with the intent to describe the core information.

The Oasis blueprints consists of these elements
OASIS methodology
The OASIS methodology is highlighting the road to recognizing and describe which services needed to realize the business goals, objectives and its necessary capability's[5].
OASIS reference model
What is a reference model? A reference model is an abstract framework for understanding and describing important entities and their relationships, an ontology. The OASIS reference model has as a primary goal to create a foundation for a SOA vocabulary .It raises the question what is a "service oriented architecture" and try to address this. The reference architecture is more concrete and as a result it takes the concepts in the reference model and expand on them.
OASIS reference architecture
What is a reference architecture? A Reference architecture describes a domain with respect to its abstract achitectural elements from a non- vendor and technology independent view. As mentioned above the OASIS reference architecture takes the reference model a bit further, and also additional concepts are introduced due to the need for addressing the core questions of the Reference Architecture.

D.1 OASIS methodology

The methodology are in the context of the OASIS SOA reference model[6] and is addressing.

- Why services needs to be defined
- How to identify the shared and supporting services
- The importance of a common language????
- How to define interactions between services at a high level
- The categorization of services for management

. but this methodology excludes

- Defining how processes work between services
- The full Enterprise or Solution Architecture
- The technical requirements of services

- The functional requirements of services
- The implementation of services
- Management of service programs

The SOA adoption blueprints is in other words a set of functional descriptions and working examples. It provides a business problem statement, a set of business requirements and a normative set of functions to be fulfilled where vendor specific details are abstracted. It is supporting the use of the OASIS SOA reference model and reference architecture, which spans over the whole Service oriented architecture.

The strategy is to use this methodology at the beginning of the development of an service architecture, to grasp the service concepts out of our preception of the domain. Only then it will become a truly service oriented they claim[5].

They follow a four step process to develop an service architecture. The four processes are What, Who, Why and How. The methodology is mostly about the three first steps and only provides a direction for the fourth. The authors explicitly say that the methodology does describe how to define the service architecture, not how it can be delivered. The first phase What is about defining a scope of the services and what they should be. Number two Who externally is driving the services, and to whom do they interact. Third is the why, why is internal and external services interact with each other. The forth and last is how, which is only given the direction for. How should they should be implemented. This is details about processes that co-ordinate services and in addition the details about how a service will be implemented. This four step process is all about starting to focus on the higher order elements first, and this will provide the context for further refinement. So according to the adoption blueprint methodology the Service architecture defines the What, identifies the Who, highlights the Why and does not do the how. The intention of this approach is that future phases will refine and detail elements discovered in the scope of the adoption blueprints. Even if the refining and detailing adds new stuff to the elements, it does not alter the defined services and their relationships. The what drives the who, the who drives the why and the why drives the how. Never the other way around. This approach does not try to describe the full amount of information that is required to deliver a complete service architecture solution or enterprise, which is deliberately left to later phases. The SOA philosophy only provides a framework and the how must be dealt with to provide a complete architecture, which is not within the scope of the OASIS adoption blueprints.

Well known architectural methodologies like Zachman[**?**] and IAF[**?**] begin the process of defining the business by investegating the context of the system or enterprise, the reason for its existence and its intentions. This is a good beginning for both the business strategy and architecture. There has been some attempts to automate the aligning of business processes and strategy to technology,but these has in the best cases been week and normally

non-existent. This is mainly because of the lack of tracability between business, architecture, delivery and management. They argue that the representation of the services must be understood by the business rather than a representation in a traditional system development tradition. By starting with a business representation, which is later expanded and refined into architectural and technical services will enable clear, auditable **trailability** between the Business Strategy through It-strategy and Enteprise Architecture to project implementations. The authors uses this definition of a service:

> "a discreet domain of control that contains a collection of tasks to achieve related goals"

This is a definition that can be used by both business and technical side, and supports the idea of bridging the gap between business and technology. In this context a domain is a distinct area with well defined boundary's. This enables a **common language** and interactions between all areas of organizations or project. In addition it creates a understanding of a common goal for distributed teams as well as on site, so that everyone is clear on what is to be delivered.

D.1.1 Big picture

One of the major goals of creating a service oriented architecture is to get the big picture. The big picture provides a overall guide to the enterprise, or a project, and will give foundation for splitting the capability's of the organizations or project into services. It will also give a deeper understanding of how change requests may be handled, and business change embraced through IT support. It is very important that all implied partys agree on the big picture because it gives all the stakeholders the context which all the details are to be related to. Namely the "what" and the "why".

The main strategy here is to start at the top of the domain. This is for several reasons. The main reasons are

- Organizations works top-down
- Reduces clutter
- Uses organizational functions as a basis.

Using organizational functions of a business as basis for identifying services is not a new idea. The new idea is to not use a process driven approach, but a service driven design. This does not give answer to all the questions that arises but gives clarity around the key questions, which is- "what services must be made available to realize the goals that the business are expected to reach?". The object is to use the organizational functions which can span across the

organizational units, the “What”. Not the temporal representation of those, within units in the organizational chart. In other words defining the functional grouping of an organizations, and hence the services that it provides, both internally and externally.

D.1.2 Collaboration

The key to create a service architecture is all about creating a common dialogue between the various different groups and defining boundaries that work across the business organizations. The best way to create a good services architecture is to use the set Collaborative working[31]. The methodology suggests that one starts with an intensive session approximately 1-3 days depending on the scope of the problem. All the stakeholders and needed information for decisions must be in place. This means people that work both outside and internally of the organizations, who understand the business functions and their relations.

By getting everyone together and agreeing on the big picture, the event creates an artifact that will be of great value for the organizations and its creation and maintenance of the service architecture. Once the decision for an architecture has been made the “Big Picture” becomes a living artifact and used in further development of the service architecture. The big picture is now object of analysis and periodically reviews to cope with change drivers both internally and externally. The reviews should normally not re-create the whole service architecture but focus on the areas change drivers impose change. Its important that these artifacts are available through collaborative environments that people can easily access for discussions and further development of the design artifact. If this issues is not addressed and the “ big picture ” is not made available, the artifact becomes just another great plan at the bottom of the drawer and will not have any value for the organizations. The approaches and terms in the service architecture must also be used across all parts of the business and enterprise architecture to ensure that solutions made or used are completely traceable. When coping with fundamental change. If reviews reveal fundamental changes to the way the business is operating the whole architecture needs to be revisited, and a new top down architecture is re-created. This usually is triggered by a large scale acquisition or merge.

To determine services, outside looking in. When determining a services architecture it is important to have people imagining that they are looking at the organizations from the outside, and describing what they see. What types of work are being undertaken. “What does it do?” is the question at this level of abstraction. At this is the level 0, the objective is to comprehend the form of the service rather than drilling down to the details. Say a employee expressed something like, “well I write down the order details, call the logistics to check inventory, if all products are available I send one copy to accounting and one copy to the logistics delivery department”. The trick is to have the organizations agree upon terms, that groups the chain of events like the one

above together. So what kind of information needs to be captured during the service definition process. It should be information at a high level and with focus on the drivers of requirements rather than the details. Their recommendation is to make a table with name, roles, a small description and its priorities. Something like this:

Name Role Description Priorities

Customer Organization or Small to large Availability privates scale retail and prize that buy products

D.1.3 Level 0

The essence when investigating architecture at level 0 is that the services discovered must be in the core and central to the organizations which is being considered. This means that support services is not a level 0. Its important to know that a level 0 service could potentially be considered as a area of its own rights, so a replacement would make minimal impact on the other services. This also means in an organizational view that this core areas could be subject for outsourcing, partnering or sale. A thumb rule they suggest is that the number of core services should be between 1 and 5.

D.1.4 Drilling down to level 1

At level 1 the services becomes more real in the sense that these can be identified not just as areas in which people work but as the actual day to day areas of concrete worktasks, and IT services that will be implemented to support them. The services becomes more down to earth, and may be mapped to the departments in the business. There are several concepts that needs to evolve when starting decomposition of services. One is that the enclosing service is giving behavior and management onto lower level services. This inheritance of principles , contracts and other functional and non-functional requirements is important for creating a well defined model. The conceptual model is like nested spheres. The second is that this classification enables navigation within the service architecture. This is to make the structure readable. To flat a structure will make the model to complex to manage, and this is one of the problems that SOA tries to solve. That is why it is so important that any service model is designed to make the problem more comprehensive and not a driver of complexity.

For this demo I have decomposed only one of the Level 0 services, namely the sales service. The sales service is then splitted up in two distinct areas. The BrowseProduct and salesOrder which is the services two big areas of sale. They interact with customer, Product services and the logistics. To help using the same terms for the same concepts, or that a change in terms are instantly reflected across the model, a modelling tool is very useful. Sometimes there is a correlation between the level 1 services and the departments, but its not in

the architectures interest that there is a deliberately one to one mapping between services and departments. These are a subject to change, and the organizations functions remains relatively complex. A service often represents the "What" but departments often represent "how".

D.1.5 Refinements, support and shared services

Depending on the gravity of the problem there might also be ?necessary? to create a deeper model with finer grained services that spans over one or several layers below level 1. Other service refinements that may be applied is to focus on the different representations that it externally might have. Services or collection of services that have a number of external actors which it interacts with.

D.1.5.1 Virtual Services Virtual services has its use when several internal services is composed to deliver a view to an external actor. Creating a virtual service does not necessary deliver a business service, nor do they incorporate any business logic, but act as a facade in front of the services. Sales or customer portals are often designed as virtual services and could be owned by the Sales level 0 service.

D.1.5.2 Support Services Support services is services that often are technology related and are quite often parted in two distinct groups. Namely technical and associated services.

D.1.5.2.1 Technical Support Services The key element of the technical support services is that they support some business function without being the business function themselves. These are normally defined at lower levels of granularity, are consumed by business services and span over multiple domains.

D.1.5.2.2 Associated Support Services The associated support services are play often an important part in the enterprise, but they are there to ensure that the business services is delivered. Associated services are often like technical services, they are shared and span over multiple domains. They should not be a part in the level 0 picture.

D.1.5.3 Shared Services As we dig deeper and the service architecture becomes more detailed, its important to identify services that are common between multiple business areas. These services may encompass support services or obvious services that are designed to work in several contexts. The degree of sharing can vary, sometimes only parts of a service are shared and in other cases the hole service is common between multiple areas. Representing shared services on different levels requires a certain level of control and

visibility. The methodology therefore suggests that shared services are categorized into two big sets. The Technical and Support services in one corner and the business services in the other. The technical and support services are then split into

- Shared between multiple business services at all levels and across service level 0 boundaries.
- Shared between multiple business services within a specific level of hierarchy
- Those with similar or common bases, but potentially differs in drivers or implementations.

The business services are split in

- Totally shared services with defined business reason for being shared.
- "Apparently" shared service, these are services that appear to have the same characteristics, but are deliberately separated.
- "Common base", these are business services that shares a common base of context, but have specialized to meet a particular requirement or business purpose.

So why these classifications? It is obvious that if we are to reuse the services that we have defined across domains, there must be some way to retrieve information about them somewhere. And a suiting class hierarchy might just be the answer both in defining what services should be a shared service and identifying those who initially seems to be a shared service candidate but really should be hosted or implemented separately. We obviously don't know how these services are to be implemented yet, but it should be possible to see the sharing potential of each service at this stage.

D.2 OASIS reference architecture

as mentioned above the OASIS reference architecture takes the reference model a bit further, and also additional concepts are introduced due to the need for addressing the core questions of the Reference Architecture. The OASIS reference architecture for SOA follows the guidelines in ANSI/IEEE std 1471-2000 recommended practice for architectural description for Software Intensive Systems.

The OASIS reference architecture aims to foster four principles. These are

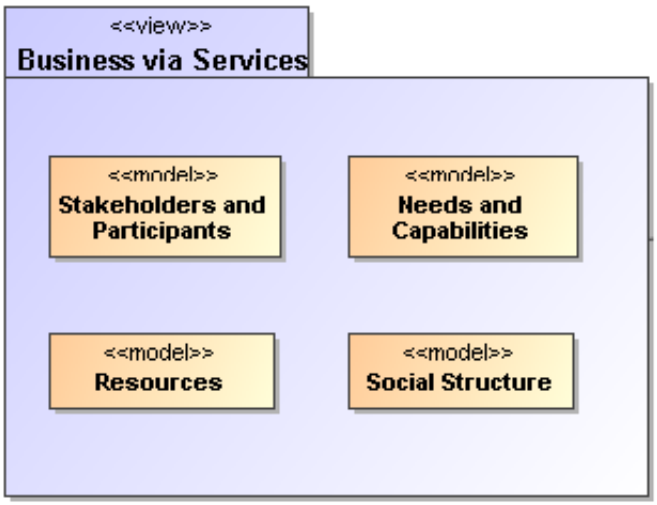
- Technology Neutrality, platform independency.

- Parsimony, keeping it simple and minimizing the number of components and their relations.
- Separation of Concerns, loose coupling and stakeholder need to know basis.
- Applicability, to cover as many aspects of the SOA as possible.

The reference architecture for the SOA ecosystem provides tree main views, first the **business via service view** that is the foundation for conducting business in the context of SOA. Secondly the Realizing Services view which addresses the detailed description of the participants, the services and its context. How is the Services realized at the platform independent modelling level. Thirdly, the owner view are addressing evolving and maintaining a Service Oriented Architecture.

D.2.1 Business via Services

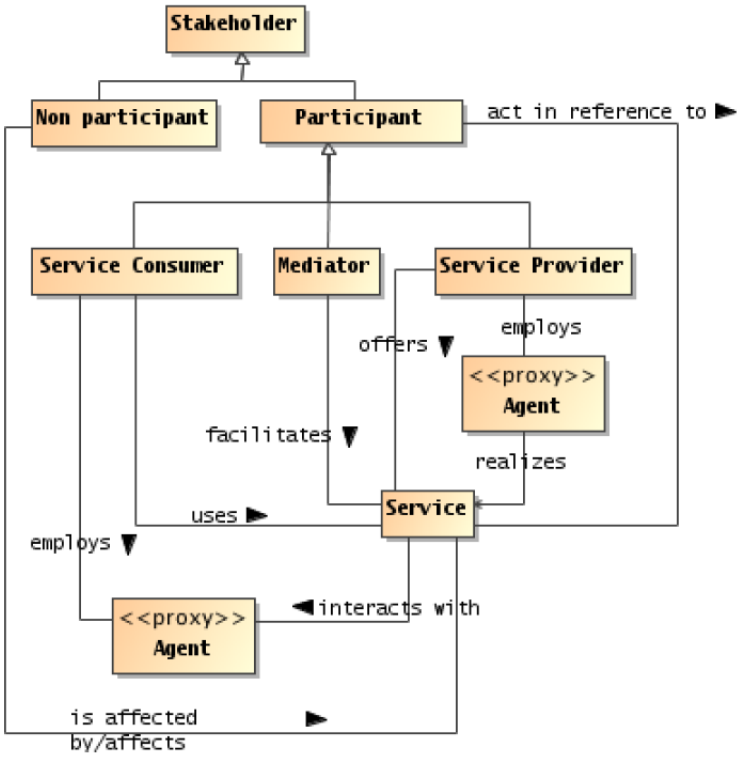
About the Business via Service view, that that has a connection to the CIM level. This view contains four elements including models for their description, and are the Stakeholder and Participant model, resources model, Needs and capability model, Social structure model and its extensions.



Stakeholders:
People,decision makers, analysts and standard architects
Concerns:

Conduct business safely and effectively
Modeling methods:
UML-class diagrams

D.2.1.1 The Stakeholder and participant model A SOA based system is existing in the context of human and non-human participants which is capable of some action. The stakeholders and participant model focus on the relationship between the users and services that they use and deploy. It is similar to other service metamodels I have seen



D.2.1.2 The Resources model In the resources model concepts like stakeholder, resource are core elements. The extension adds responsibility, right, policy and ownership. See figure 64

D.2.1.3 The Needs and capability's model The needs and capability model is describing how a service provider may use resources as the capability

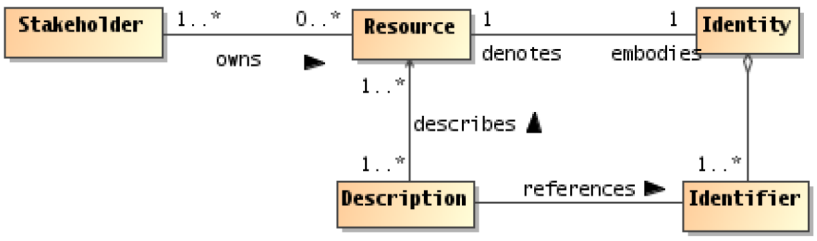


Figure 64: The resource model

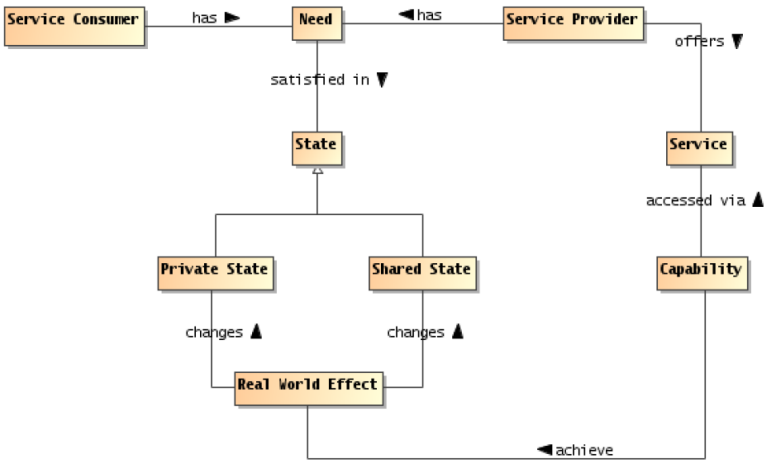


Figure 65: The Need and Capability model

to achieve realworld effects on the consumers behalf. The need is a measurable requirement that a service participant is reaching for. See figure 65

D.2.2 The Realizing SOA view

Stakeholders: Enteprise architects, business analyst, standard architect and decision makers
Concerns: Effective construction of SOA-based systems
Modeling methods: UML class, sequence, component and composite structure diagram

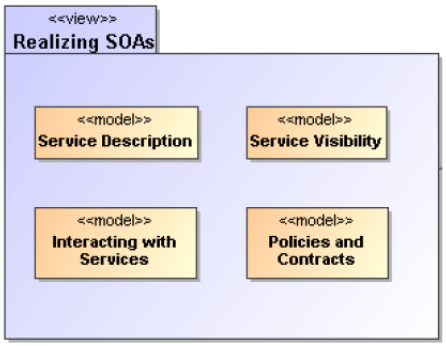


Figure 66: The Realizing SOA view

The realizing SOA view defines or describes the information needed to use, build, deploy, manage and manipulate a service. In addition to information and behavior models used to define the service interface. The description also includes information to decide if the service is fitted for service consumers needs. Information describing service reachability, service functionality, contracts and policies is also important model elements in this picture.

As seen in the two diagrams above, they are connected through the service description concept which is a subclass of the general description class. The first diagram is the general service description diagram that includes consumer and provider descriptions in addition to the service description. The Categorization concept provides a key word for making service searchable and classification for classifying the service. The Service Description Model adds concepts that are more SOA related. Among these are Service reachability, service interface, Service functionality, policies and contracts. This will enable modelling of interaction, behavior, defining functions and the identification of known or missing contracts and policies.

D.2.2.1 Service Visibility Achieving visibility is seen as one of the key success factors for the services to make a meaningful interaction and achieve awareness,willingness and reachability. The core concept in this relation is the mediator concept in between service provider and service consumer. Within the mediator concepts lies the capability to manage publishing, discovery and connection of services. It offers a mediated service awareness.

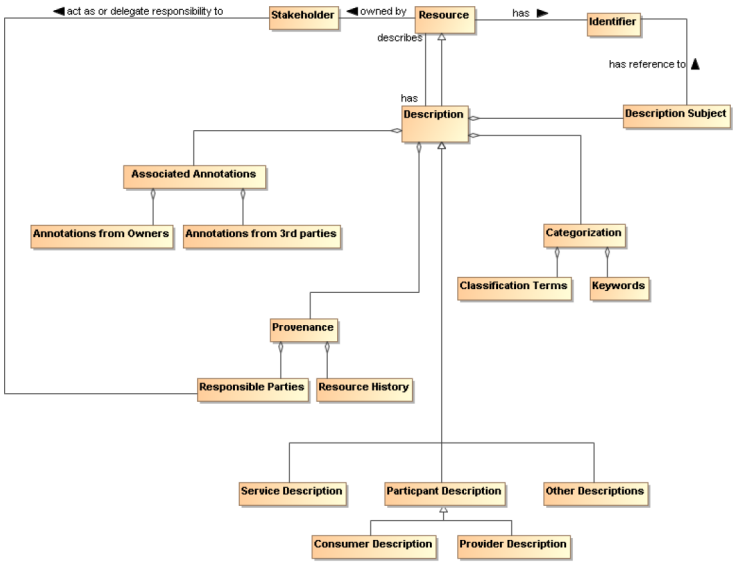


Figure 67: General service description

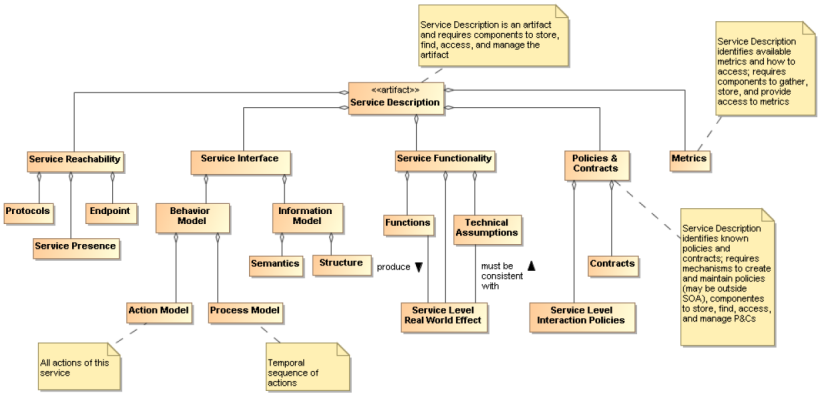


Figure 68: Service description model

D.2.2.2 Interacting with services model In the interacting with services model has a participant that receives or send an message which realizes an event or an action. The event reports of a realworld effect or an action that causes a realworld effect.

D.2.2.2.1 Message The use of a service requires use of some action, this usually is done by a series of information exchanges or altering a shared state of a resource. A message including descriptions of syntax and semantics is commonly used for this purpose. The Reference architecture also contains the message exchange patterns Send/receive and notification.

D.2.2.2.2 Composition of services The composition of service is using one or more services to compose a new service. The services are either an atomic service or composite service. Both are visible to service consumer through a single interface, with a single service description.

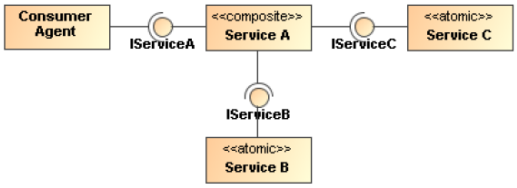


Figure 69: simple service composition

The composite services however has a service description that is an aggregation of one or more other services. These services may be atomic, a composite or both. See figure D.2.2.2.2

D.2.2.2.3 Service oriented Business process The concept of business processes and collaboration in was modelled as part of the Business via services view in the context of transactions and exchange across organizational boundaries. In the realization view the principle of composition of services is applied to business processes and collaborations. A principle that can be used to compose services is orchestration. See fig 70

Orchestration

> *A technique used to compose hierarchical and self-contained service-oriented business processes that are executed and coordinated by a single agent acting in a conductor role.[7]*

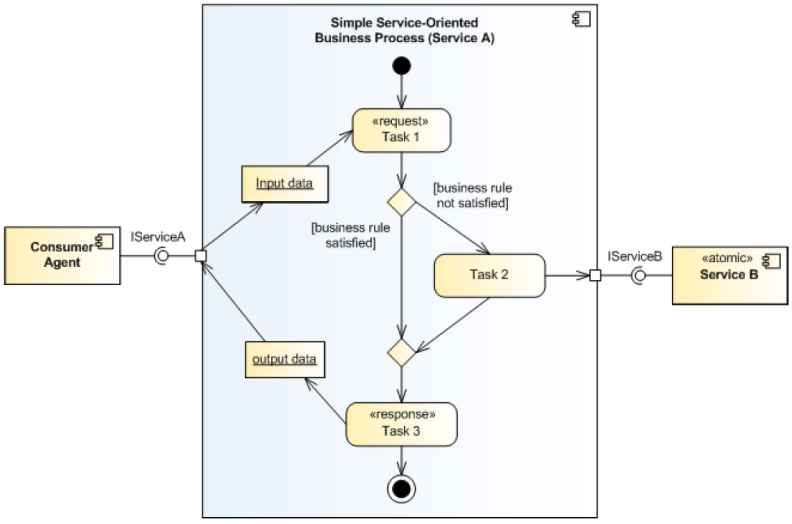


Figure 70: Orchestration of services

Orchestration is not to be confused with choreography. They differ in the sense that orchestration is telling the surroundings how to behave whereas in choreography every party is independent entities that behave according to a arranged pattern. In this particular context the entities are processes within services or agents. See Figure 71 for a generic example of choreography.

Choreography

> *A technique used to characterize and to compose service-oriented business collaborations based on ordered message exchanges between peer entities in order to achieve a common business goal.[7]*

Choreography in this context is not equal to the term service choreography used in Business via service view within the Exchange and transaction model. Which is defined as the description of the possible interactions that may take place between two or more participants to fulfill an objective. .

D.2.3 Owning SOA view

Owning SOA view is about the different aspects of owning a SOA. A SOA based system is in a living and changing world and the environment the system is a part of is a "ecosystem" in a sense. To make the system adapt to

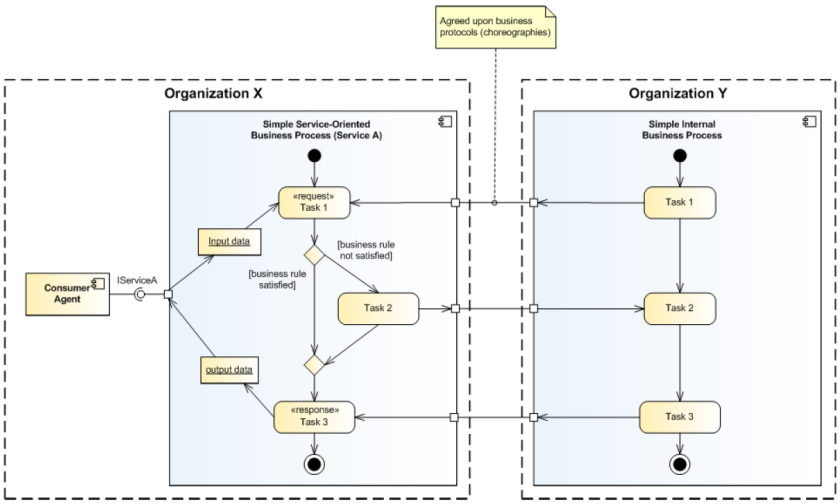


Figure 71: Choreograpy of business service processes

the ecosystem, some management and governance are needed to ensure that all its components pull in the same direction.

Stakeholders: Decision makers,Service providers and Service consumers.
Concerns: Processes for engaging in a SOA are effective, fair and assured.
modelling techniquesUML class diagrams

This view focuses on as seen in figure 72 three aspects of managing and governing SOA: security, governance and Services as managed entities.

D.2.3.1 Governance model Governance is mostly about making decisions. These decisions must be in aligned with the SOA strategy and the domains culture. Governance is important for the evolving SOA and the changing ecosystem. Setting policies for decisions that is resistible for change is a success factor in such environments. Ensuring that the services continues to fulfill the goals of the business. The generic model of governance consists of four different parts: Motivating governance(see figure 73, setting up, carrying out and Ensuring governance compliance.

For SOA to work over ownership boundaries some considerations must be made argues the authors. For instance that there is multiple governance chains and as many goals. The consumer of a service may have a policy to achieve

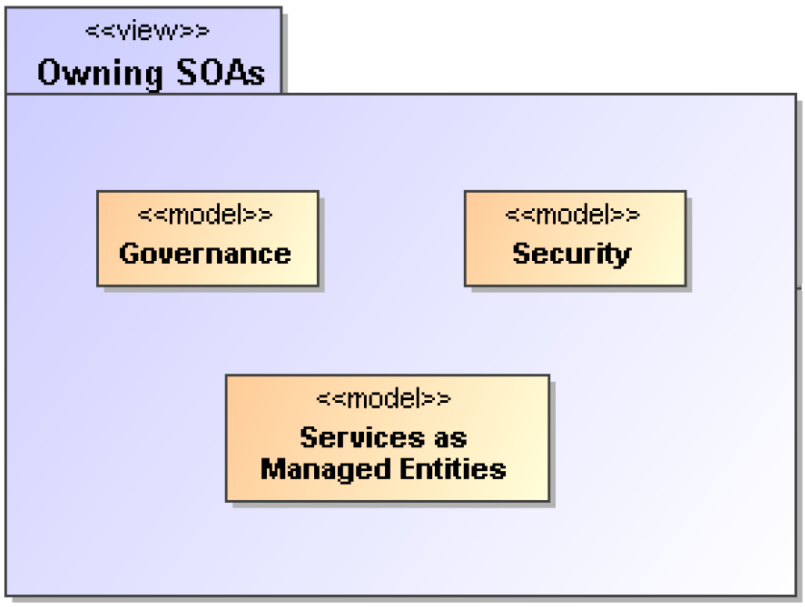


Figure 72: Owning SOA view

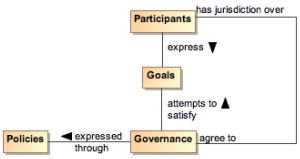


Figure 73: The Motivation governance model

the lowest possible price for a certain product but the provider wants to get the highest possible price. This becomes a challenge when the policies cross ownership boundaries and are overlapping.

D.2.3.2 Security Security is mainly about focusing and striving against confidence in SOA system. The area of security could be captured by the core security concepts or goals, which is not absolute. 100% confidentiality is not a realistic goal.

- Integrity
- Confidentiality
- Availability

- Authentication
- Authorization
- Non-repudiation

D.2.3.2.1 Trust model Trust with respect to security is the confidence participants in a architecture have when they are interacting with each other. The trust model intends to model trust in terms of participant and the authorization to perform an action. See figure 74.

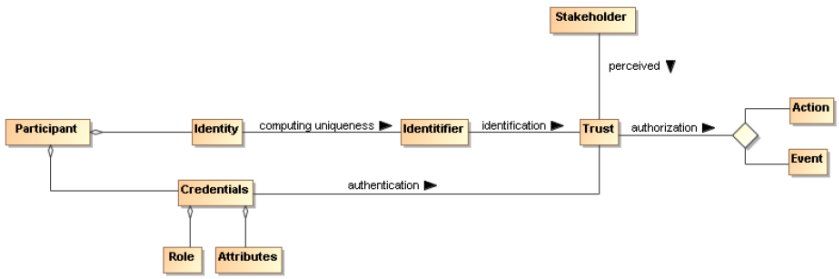


Figure 74: The Trust model

The trust model intends to describe the domains of trust. See figure 75.

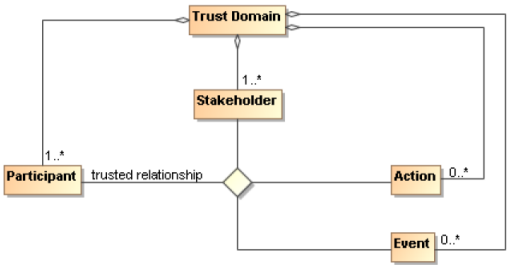


Figure 75: The Domain Trust model

The abstract space of actions that shares a common trust requirement is referred to as the concept trust domain. The kinds of trust domains can vary. In a infrastructure aspect, it could be at the networking layer for regulating

communication and access to resources. In a application context a trust domain is a within social structure with participants that have already established a trust relationship. In the area of trust the reference architecture talks about centralized and decentralized trust authority. Centralized trust authority is when a central trust administration regulates the trust domain according to given security polices. Decentralized trust authority gives more authority to the participants. The authority is delegated. The reference architecture also provide templates for policy mechanisms for security.

D.2.3.2.2 Security layers The security layer model consists of three layers of abstraction. These are the Application layer, Transport layer and Network layer

D.2.3.2.3 Threat model The threat model highlights the most usual threats to the core security goals. some of these are message alteration, message interception, man in the middle, spoofing,denial of service attack, replay attack and false Repudiation

D.2.3.2.4 Security response model The security response model describes different strategies to protect the system against threats. This could typically be by encryption or digital signatures. The security response model also brings other concepts to the table as a response to the threats. This is concepts like Privacy enforcement, integrity protection, Message replay protection, auditing and logging and graduated engagement.

D.2.4 OASIS RA summary viewpoints

Part XVI
Zachman

Below you can see the Zachman framwork matrix which has been an inspiration for many methodologies and frameworks within the EA and SOA and BPM communities. See figure 77.

Viewpoint Element	Viewpoint		
	Business via Services	*Realizing Service Oriented Architectures*	*Owning Service Oriented Architectures*
Main concepts	Captures what SOA means for people using it to conduct business.	Deals with the requirements for constructing a SOA.	Addresses issues involved in owning and managing a SOA.
Stakeholders	People (using SOA), Decision Makers, Enterprise Architects, Standards Architects and Analysts.	Standards Architects, Enterprise Architects, Business Analysts, Decision Makers, Standards Architects and Analysts.	Service Providers, Service Consumers, Decision Makers.
Concerns	Conduct business safely[8] and effectively.	Effective construction of SOA-based systems.	Processes for engaging in a SOA are effective, equitable, and assured.
Modeling Techniques	UML class diagrams	UML class and sequence diagrams, component and composite structure diagrams	UML class diagrams

Figure 76: The OASIS viewpoints

Part XVII

POSI extra

This part is for showing pictures the dont nesseseary fit in between text. Below is the evolving SOA figure. The participants out of the picture is Financial institution on top and Shipper below in figure 78.

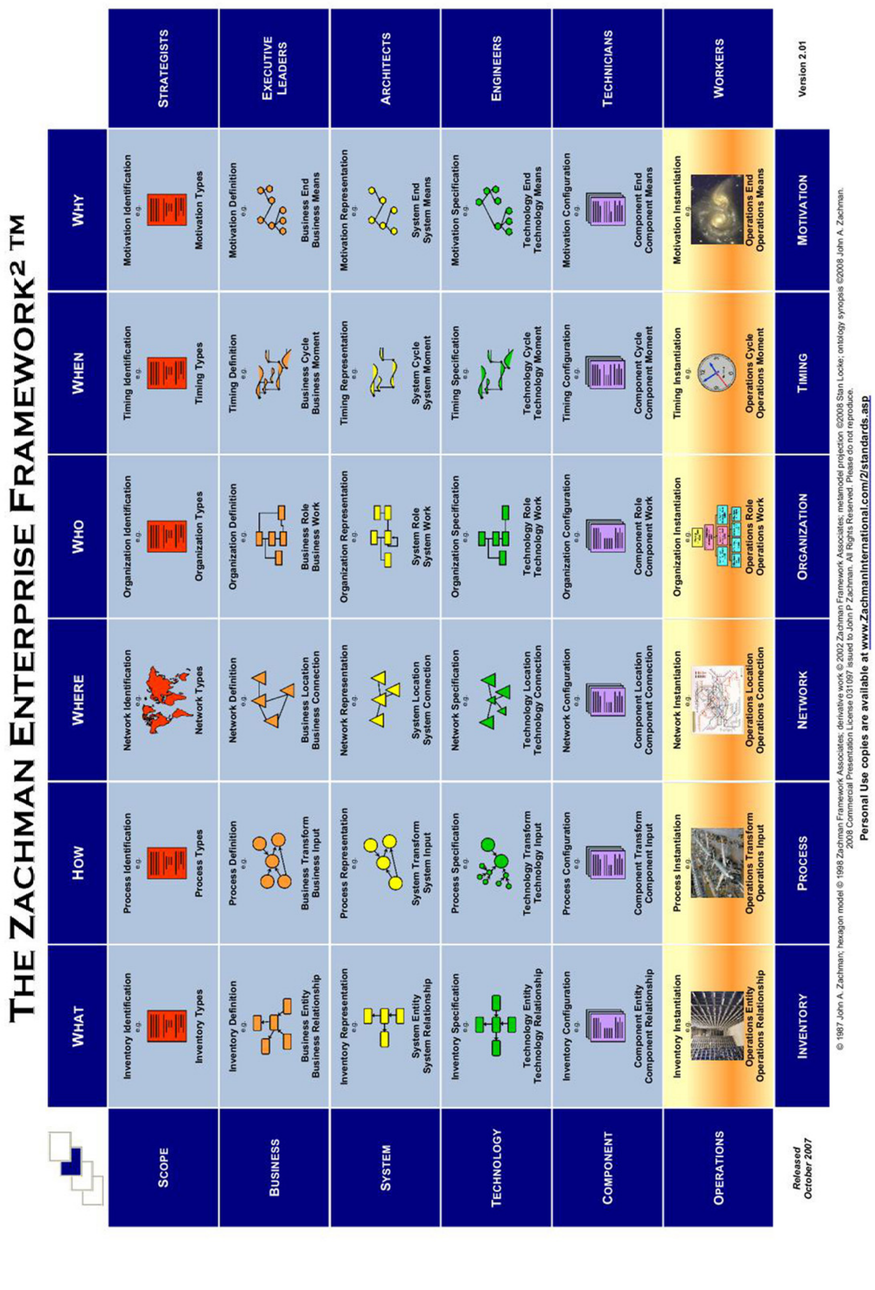


Figure 77: Zachmann Framework

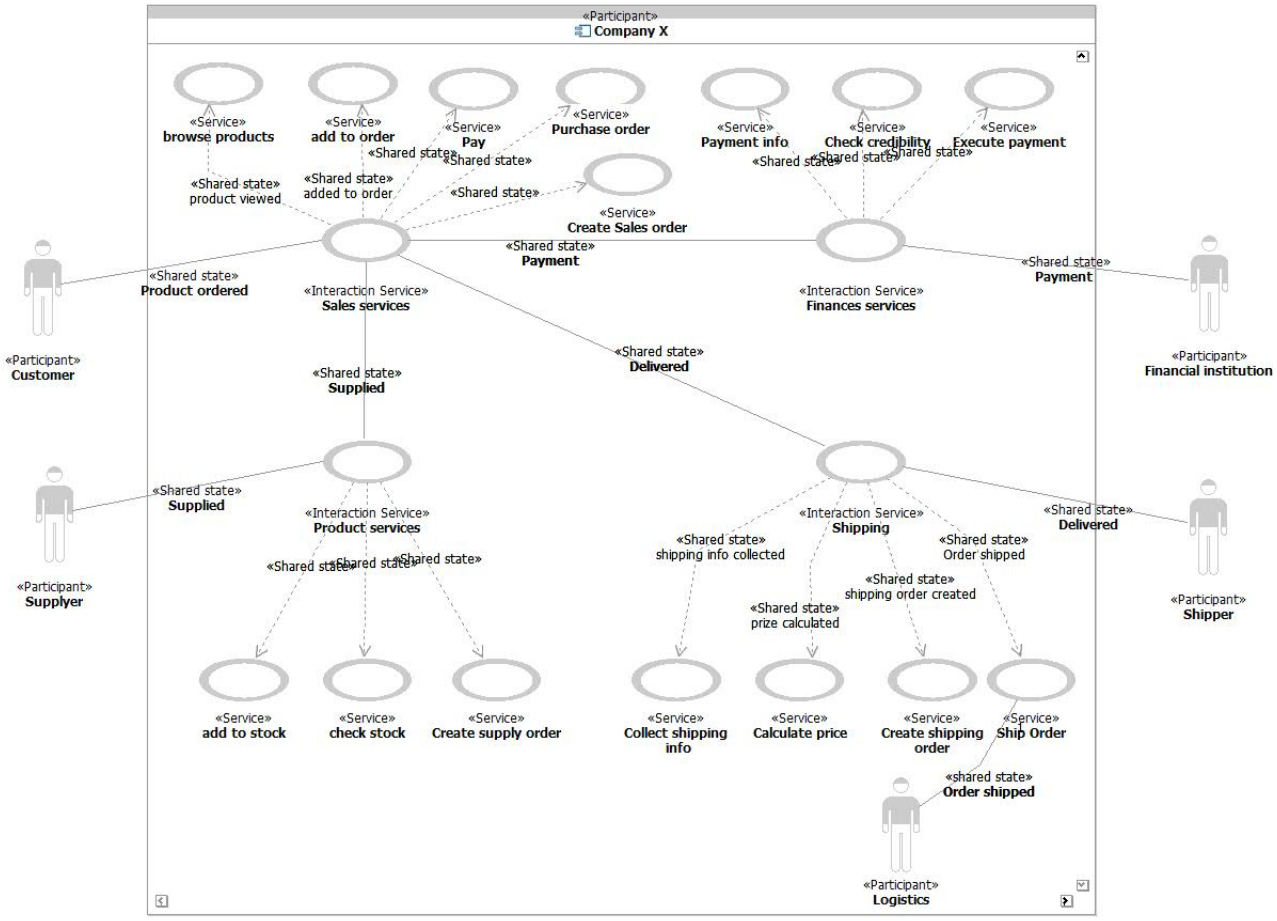


Figure 78: The evolving SOA